2007

Computational aspects of voting: a literature survey

Fatima Talib

Fatima Al-Raisi
May 16^{th}, 2007

Abstract

Preference aggregation is a topic of study in different fields such as philosophy, mathematics, economics and political science. Recently, computational aspects of preference aggregation have gained especial attention and "computational politics" has emerged as a marked line of research in computer science with a clear concentration on voting protocols. The field of voting systems, rooted in social choice theory, has expanded notably in both depth and breadth in the last few decades. A significant amount of this growth comes from studies concerning the computational aspects of voting systems.
This thesis comprehensively reviews the work on voting systems (from a computing perspective) by listing, classifying and comparing the results obtained by different researchers in the field. This survey covers a wide range of new and historical results yet provides a profound commentary on related work as individual studies and in relation to other related work and to the field in general. The deliverables serve as an overview where students and novice researchers in the field can start and also as a depository that can be referred to when searching for specific results. A comprehensive literature survey of the computational aspects of voting is a task that has not been undertaken yet and is initially realized here. Part of this research was dedicated to creating a web-depository that contains material and references related to the topic based on the survey. The purpose was to create a dynamic version of the survey that can be updated with latest findings and as an online practical reference.

Preface

The notions of preference and preference-aggregation are fundamental notions in many disciplines such as philosophy, social choice theory, mathematical decision theory, economics, applied mathematics, and computer science. In computer science, the topics of "preference" and "preference aggregation" are studied in Artificial Intelligence for coordinating a society of software agents, in Databases for filtering the information retrieved in response to queries, in Natural Language Processing for classifying different tokens in a text, and in Complexity Theory for analyzing the computational complexity of various preference aggregation methods.

Among Preference aggregation methods is the naturally appealing method of voting. Recently, computational aspects of preference aggregation through voting have gained especial attention and "computational politics" has emerged as a marked line of research in computer science with a clear concentration on voting protocols. The field of voting systems, rooted in social choice theory, has expanded notably in both depth and breadth in the last few decades. A significant amount of this growth comes from studies concerning the computational aspects of voting systems.

This thesis extensively reviews the work on voting systems (from a computing perspective) by listing, classifying and comparing the results obtained by the researchers in the field. This survey covers a wide range of new and historical results yet provides a commentary on related work as individual studies and in relation to other related work and to the field in general. This work serves as an overview where students and novice researchers in the field can start and also as a depository that can be referred to when searching for specific results. Each chapter is concluded with comments and bibliographic notes and is appended with separate bibliography. The entire thesis is appended with a comprehensive index of subjects/terms and an index of authors that contains the names of over two hundred authors of works in computational aspects of voting cited here[1]. At the end is a list of all references sorted in alphabetical order by author (last) name. The list of references contains more than two hundred items including books, book chapters, conference proceedings, journal articles, technical reports, and other forms of published and unpublished work. A comprehensive literature survey of the computational aspects of voting is a task that has not been undertaken yet and is initially realized here.

[1] Names of authors of cited works appear in the authors index whether or not the names are explicitly mentioned in the text.

Part of this research was dedicated to creating a web-depository that contains material and references related to the topic of the survey. The web-depository is a dynamic version of the survey that can be updated with latest findings and used as an online practical reference. Both the URL of the dynamic survey and the electronic copy of this thesis can be accessed at: `http://www.cs.rit.edu:8080/ms/static/index.html`.

Acknowledgements

He who does not thank people is not grateful to God.

Prophet Muhammad (pbuh)

Most of all, I would like to thank my advisor, and teacher, Edith Hemaspaandra. I thank Edith for triggering my initial interest in Computational Politics and in Complexity Theory in general. Throughout my thesis development, Edith has been a great advisor and teacher. She encouraged me to work in this area, provided advice and guidance, promptly responded to my queries, and taught me countless things from the basic concepts to the details of structuring and presenting the work. I am grateful for the opportunity of being Edith's advisee and student.

I also thank the other members of my committee, Chris Homan and Piotr Faliszewski. I thank Chris for agreeing to serve on the committee on a short notice, and then for his valuable comments on the text. His feedback has contributed to improved quality of this work. Special thanks go to Piotr Faliszewski for pointing out some related works that I was not aware of, and for clarifying some points.

I have been very fortunate to have such a great committee.

I would also like to extend my thanks to Stanislaw Radziszowski for insightful discussions on the recommended depth of the chapter on Electronic Voting.

I must acknowledge as well my colleagues at RIT, Nathan Russell and Eric Brelsford for providing information and answering questions regarding their theses, so that I can include their work in the survey, and my colleagues and friends at both RIT and SQU (Sultan Qaboos University, Oman) for reading various parts of this relatively long Master's thesis: Jeremy Paskali, Fatou Bagayogo, Youcef Baghdadi, and especially Anisa Al-Hafeedh.

I am always indebted to the people whom without I wouldn't be where I am and who I am today, my family. I thank my family, and especially my parents, for their support, love, and patience while I was thousands of miles away pursuing my Masters in Computer Science.

Finally, to God is all praise and ultimate thanks.

Contents

List of Tables

List of Figures

Chapter 1

Preliminaries

In seeking private interests, we fail to secure greater collective interests. The narrow rationality of self-interest that can benefit us all in market exchange can also prevent us from succeeding in collective endeavors.

Russell Hardin (Collective Actions)

1.0 Prologue

This chapter establishes a foundation needed to read most of the following chapters. It also provides an overview of the field and touches on some aspects that later chapters will discuss in more depth.

Some mathematical concepts will be used throughout the chapters of this thesis. Therefore, the first section of this introductory chapter defines these concepts. A reader who is familiar with the basics of set theory and data structures may skip the *Mathematical Preliminaries* section. The second section introduces *Social Choice Theory* and serves as a link to the third section which is on the preliminaries of *Elections and Voting Systems*. Examples and properties of voting systems are given in the fourth and fifth sections respectively: *Some Voting Rules* and *Properties of Voting Rules*.

As an introductory and a preparatory chapter, it has an emphasis on terminology and definitions with some few examples. The last section, however, concludes with *Comments and Bibliographic Notes*.

1.1 Mathematical Preliminaries

1.1.1 Elementary Concepts of Set Theory

A set is an unordered collection of objects called elements and these elements must not be repeated, in other words, a set contains distinct elements. A list, however, contains ordered elements. A multiset may contain repeated elements where order is not important. In a list, both order and multiplicity are significant whereas in a multiset multiplicity is significant but order is not. In sets, both are ignored.

Element x belongs to set S is denoted $x \in S$. A subset of a set S is a set whose elements are contained in the set S (i.e., each of them belongs to S). Set A is a subset of S is denoted $A \subseteq S$. If A is a subset of S then S is a superset of A by definition, and this is written $S \supseteq A$. A proper (or a strict) subset of a set S is one that does not contain all the elements of S. A is a proper (or a strict) subset of S is denoted $A \subsetneq S$. The empty set $\emptyset$ is the set that contains no elements and it is a subset of any set.

The cardinality of a set S is equal to the number of elements in S and is denoted by $\|S\|$ (or $|S|$, but since this notation is used to denote the absolute value of the argument, when it is a number, we will avoid using it to stand for the cardinality of a set). The intersection of two sets A and B, denoted $A \cap B$, is the set of elements that are common to both A and B. The union of two sets A and B, denoted $A \cup B$, is the set that contains all the elements of A and B together. The complement of a set S, denoted S', is the set that contains all the elements not in S (with respect to a defined universal set that contains all the elements in the discussion's *universe* of interest).

A partition of a set S is a set that contains subsets of S such that the intersection of any two of these subsets is empty and the union of these subsets is equal to S. The Cartesian product of two sets A and B, denoted $A \times B$, is the set of all ordered pairs (x, y) where x is an element of A and y is an element of B. A binary relation R on a set S is a subset of the Cartesian product of S with itself (S^2). xRy is an alternative expression to $(x, y) \in R$.

1.1.2 Some Data Structures

Graphs

A graph is simply a collection of points and lines. Formally, a graph is a tuple (V, E) where V is the set of vertices or nodes (the points) and E is the set of edges (the lines) each connecting some pair of vertices $x, y \in V$ and is represented by an ordered pair (x, y) where $x, y \in V$ or simply by a collection or a set of two elements $\{x, y\}$. Different types of graphs exist mostly depending on constraints on the set of vertices or edges. In this simple common type of graphs, at most one edge (i.e., either one edge or no edges) may connect any two vertices.

If the edges are directed, i.e., start in one vertex and end in another (or point to

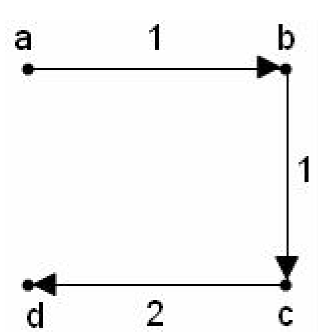

Figure 1.1: A weighted directed graph.

another), then the graph is a *directed* graph and edges in this case must be represented by ordered pairs in which the first element indicates the source vertex and the second element indicates the destination vertex. The direction of an edge is visually represented by an arrow on one of the endpoints, the destination vertex. In a directed graph, directed edges are sometimes referred to as arcs.

A cycle in a graph is a series of edges such that the source vertex of the first edge coincides with the destination vertex of the last edge, in other words, the cycle edges form a cycle by starting and ending in the same vertex. If edges are associated with numeric values then the graph is a *weighted* graph and the weight of an edge is the number associated with it. Figure 1.1 shows an example of a weighted directed graph where $V = \{a, b, c, d\}$ and $E = \{(a, b), (b, c), (c, d)\}$. The numeric labels on edges are the weights.

Binary Trees

A tree is a special graph that is connected and acyclic, in other words, there are no cycles in a tree and every vertex is connected to every other vertex through some path. A unique vertex is called the *root* of the tree, trees which such a designated node are also called rooted trees.

A vertex y that is one edge away from a given vertex x, with x being closer to the root, is called a *child* of x and x is called a *parent* node to y.

Vertices connected to the same parent node that are the same distance from the root node are called *siblings*. The root vertex has no parent and *leaf* nodes have no children.

A binary tree is a tree in which every vertex has at most two children. The relative position of a child to its parent vertex is significant in binary trees; each child of a vertex is designated as its left or right child. A binary tree is balanced if and only if the difference between the maximum and the minimum depth among the leaves is less than or equal to 1. Figure 1.2 depicts an example of a binary tree (which is also balanced since all leaves have depth 2).

1.1.3 Binary Relations and Rankings

A binary (i.e., consisting of two parts) relation is complete if and only if for any two distinct elements x and y, either xRy or yRx (or both).

A binary relation is reflexive if and only if for all elements x in the set, xRx. It is irreflexive if and only if for all elements x in the set, it is not the case that xRx (this is written $\neg(xRx)$ and it stands for the negation of xRx). Note that "irreflexive" is not the same as "not reflexive," in other words, one is not the logical negation of the other.

A relation R is symmetric if and only if xRy implies yRx for all elements x and y in the set. A relation R is asymmetric if and only if xRy implies $\neg(yRx)$ for $x \neq y$. Here also, the properties symmetric and asymmetric are not logical negations of each other.

A transitive relation R is one that satisfies the following condition. For all elements x, y, and z in the set: if xRy and yRz then xRz. If only this condition is satisfied then the relation is transitive, in other words, this condition is both sufficient and necessary for transitivity. A relation is negatively transitive if and only if for all elements x, y, and z in

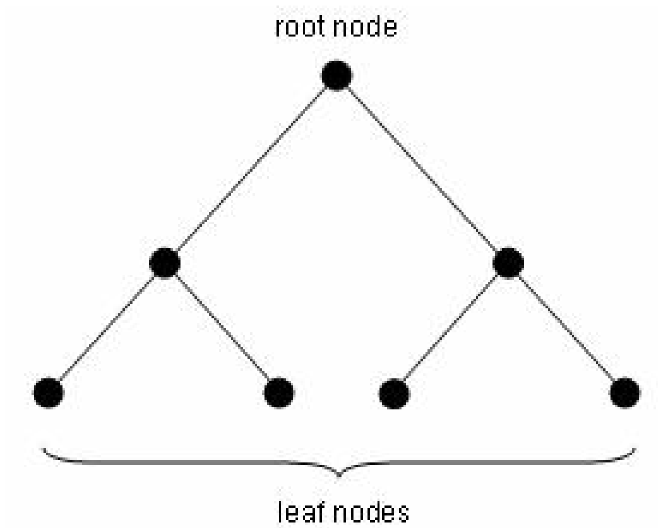

Figure 1.2: A binary tree.

the set: $\neg(xRy)$ and $\neg(yRz)$ implies $\neg(xRz)$.

A binary relation is called a strict ordering or an irreflexive ordering if and only if it is complete, irreflexive, asymmetric, and transitive. If the condition of irreflexivity is relaxed then a complete, asymmetric, transitive, and negatively transitive binary relation is called an ordering. An ordering is also referred to as a *ranking* in the context of elections. In that context, the set of elements is called the candidate or alternative set. Implicitly, we assume the existence of a voter set. The voters cast their votes which are a type of ordering on the elements of the alternative (or candidate) set. The following sections employ the concepts mentioned above in the voting/election context.

1.2 Social Choice Theory - an Introduction

As social choice theory preceded computational politics (and computational voting theory in particular), some concepts in computational voting theory stem from former concepts in social choice theory. Therefore, we start defining the basic terms in computational voting theory by explaining the counterparts in social choice theory.

Social Choice Theory is a field of study that explores and investigates procedures for *aggregating* individual choices in a society and producing a *social* choice that is based on the choices of the members of the society. The theory of social choice is interested in fair and uniform protocols for group decision-making. Group decision-making essentially involves aggregation of individual preferences into one congregational outcome that best reflects the individual choices and is the closest possible to a consensus opinion or is at least a globally desirable outcome. In this context, a choice of an individual is viewed as a binary relation that compares two alternatives. When there are more than two alternatives, a choice is expressed in terms of a number of binary relations each expressing the choice between a pair of alternatives.

This setting implies a set of alternatives and choices are made to express the preferences of individuals over the set of alternatives. These choices are expressed in terms of a binary relation, which is mathematically defined as a subset of the Cartesian product of two sets; in this case the set of alternatives and itself. This binary relation may be a strict preference relation or a weak preference relation. A strict preference relation expression, denoted xPy for the society and xP_iy for an individual i, means that alternative x is preferred to alternative y but alternative y is not preferred to alternative x. A weak preference relation, denoted xRy for the society and xR_iy for an individual i, implies that x is considered to be as good as or better than y, or in fewer words: x is no worse than y. This can also be denoted xPy or xIy where I stands for an indifference relation. In real life scenarios, it is not unusual to have choices that are indifferent between two alternatives. This is the reason for using a weak preference relation; it takes into account the possibility of indifference.

The weak preference relation R has a number of properties: completeness, symmetry, and transitivity. R is a complete relation because for any two distinct alternatives, x and y, either xRy or yRx (or both). It is also symmetric, xRx, since any alternative is as good as itself. A property of a preference relation (at least for individual choice before preferences are aggregated) is transitivity. This property states that if xRy and yRz then it must be the case that xRz. It is irrational to describe x as no worse than y and y as no worse than z but then describe x as neither as good as z nor better than z. This means that x is worse than z. If we think of the weak preference binary relation as similar to the greater than or equal relational operator on real numbers then the above irrational claim is similar to saying that $x \geq y$, and $y \geq z$ but $x < z$ where x, y, and z are real numbers. Transitivity is therefore the hallmark of rationality. Individual choices are expected to be rational; they are expected to be transitive. This applies to the strict preference relation P as well. In addition, P is also complete but it is asymmetric, so for any alternative x, x cannot be preferred to or better than itself, this is denoted $\neg(xPx)$.

Interestingly, individual rational choices do not necessarily imply a rational social choice! This observation was first made by Condorcet in his analysis of the majority rule. Condorcet noticed that when some transitive individual preferences are aggregated (using the pairwise majority rule), the result is not a transitive preference. For example, consider the following preferences of three voters $\{i, j, k\}$ over a set of three alternatives $\{x, y, z\}$:
Voter i's preferences are $x > y > z$, voter j's preferences are $y > z > x$, and voter k's preferences are $z > x > y$.

Condorcet noticed that alternative x is preferred to y by a majority of votes (two votes—from voters i and k—out of three), alternative y is preferred to z by a majority of votes, but alternative z is preferred to x by a majority of votes also which clearly clashed with transitivity since by transitivity x should be preferred by z in the aggregate vote. This irrationality phenomenon in the majority rule is some referred to as the paradox of "cyclic majority." The cycle in the previous example can be seen by writing the aggregate preference as $x > y > z > x$.

Irrationality is a multifaceted phenomenon in vote aggregation. Intransitivity is one of them. There are other voting paradoxes in addition to the Condorcet paradox explained above. The conclusion is as before: rational individual choices do not imply rational collective choice. This remarkable result was shown—to apply in general and not only in the case of the majority rule—by the pioneer of modern social choice theory and Nobel Prize winner Kenneth Arrow [1]. This also draws attention to the importance of the way the individual choices are transformed into a social choice. This is the role of a social preference function (also called social choice procedure, and social welfare function, but the latter is rather an obsolete term today).

A social preference function takes as input a collection of individual preference relations, R^n, that contains a preference relation for each individual in the society. This collection of individual preference relations is called a social preference profile $R^n = (R_1, R_2, \ldots, R_n)$ where n is the number of individuals in the society and R_j stands for the preference relation of individual j in the society.

The social preference function takes as input a social preference profile, R^n, and outputs a social preference relation R. A social *choice* function outputs only the most preferred outcome (among the input alternatives) and hence is not informative about the position of other alternatives in the final outcome. Both a social preference function and a social choice function can be seen as special cases of a social preference correspondence which produces a *non-empty subset* of outcomes (whether an outcome is a full ranking or is simply a subset of alternatives). In the case of a social preference function, it produces a singleton: a set of one preference relation. The term social *welfare* function stands for the same concept as social preference function, but it is preferred not to use this term[1]. The domain of a social preference function is the set of all possible social preference profiles which is the n-times Cartesian product of $\mathcal{R}$, where $\mathcal{R}$ is the set of all possible weak preference relations. This Cartesian product is denoted $\mathcal{R}^n$. Recall that weak preference relations are used instead of strict preference relation mainly to take indifference choices into account. However, this assumption of using weak preference relations to represent individual choices may vary according to the model, scenario and intended goal of the discussion.

The range of social preference function is, therefore, the set of all possible weak preference relations $\mathcal{R}$. The few sentences above are mathematically written as $F : \mathcal{R}^n \to \mathcal{R}$, where F stands for a social choice function. Arrow's possibilibly theorem intended to investigate the *possibility* of defining or coming up with a social choice function under conditions that he laid out as necessary and are thought of as the minimum requirements for fairness and uniformity. Arrow postulated that those are reasonable pre-requisites and formulated them in the following conditions:

1. Universal Domain which means that the domain should not be restricted to a subset of all possible social preference profiles, that is, the individuals in the society are free to make any choices as long as they are rational (complete and transitive).

[1] It was pointed out by Johnson that it is better not to use this term because it "confused some readers and carried emotional baggage for others" [14].

2. Nondictatorship: the choice of any one individual alone (with no regard to the choices of other individuals in the society) must not solely decide the outcome of the function.

3. Pareto efficiency: this means that the social choice function must be consistent with the unanimous choice: if all individuals make the same choice (e.g., prefer alternative a to b) then the social preference relation must have the same choice (prefer a to b in the previous example).

4. Independence of irrelevant alternatives: if two social preference profiles agree on a subset of alternatives (e.g., all individuals rank a over b) then the corresponding social preference relations should reflect the same choice in both cases (e.g., rank a over b) regardless of the place of other *irrelevant* alternatives in the preference profiles, that is, the social preference relation between two alternatives, say a and b, is not affected by the position of some other irrelevant alternative, c.

Starting from these conditions or axioms and proceeding with logical reasoning, Arrow showed that a social preference function does not exist. In particular, he showed that if a social preference function satisfies the universal domain, unanimity, and independence of irrelevant alternatives conditions then it must violate nondictatorship. Since that important result, Arrow's theorem is often referred to Arrow's *impossibility* theorem.
We remark that Arrow's theorem holds (as a theorem) *given* the conditions preceding the proof. The proof is easily perceived as sound and *logical*, in technical terms, once we accept the conditions of the Arrovian framework as *axioms*. Arrow's theorem has drawn wide attention and studies have been dedicated to investigating the conditions of the theorem, its theoretical bearings and practical implications.

Interested readers should refer to Arrow's original work [1] or to other work that explain the axioms, the theorem, its proof, and the implications of this notable result, for example [3, 25] and [14, 9].

1.3 Elections and Voting Systems

1.3.1 Definitions and Terminology

An election $\mathcal{E}(C, V)$ consists of a candidate set C and a voter set V. For the remainder of this thesis, the set of candidates C and the set of voters V will be finite sets[2]. There are different representation issues related to the input of the candidate and voter sets. The assumptions and representations vary from one work to another. The set of candidates is implicitly drawn from the votes in some cases and is assumed to be explicitly input in some others. The voter set is sometimes explicitly input and in some treatments it is identified by the list of votes (where each vote in the list represents a voter). In any case, most related research work—specially that concerning complexity-theoretic aspects of voting systems—assume that the votes are strict ordering over the alternatives. It is worth mentioning here that orderings or qualitative representations in general are not the only form for expressing

[2]Nevertheless, in other contexts the set of candidates/voters is continuous.

choices and preferences, quantitative and hybrid representations are also used to declare preferences. Chapter 3 of the thesis will cover representation issues in more detail.

As an introduction, an election consists of a candidate set and a voter set. An election system or a voting system (interchangeably) is a system that inputs the candidate and voter sets and outputs a winner set[3] according to a well-defined rule that ought to be fair and uniform. However, with knowledge of Arrow's impossibility theorem, this implies the assumption of relaxing at least one of the conditions or desirable properties of a social preference function (in this context: an election system or a voting system).

Although voting systems are similar to social choice functions, they are different in one clear aspect: voting or election systems produce a winner set from votes whereas social choice functions produce a ranking of all alternatives involved in the voting procedure. The winner set is a subset of the candidate set and it is the set of top-ranked alternative(s). In some scenarios, a unique winner is sought and so the outcome of the election is a single member of the candidate set. In some cases, however, more than one winner is possible through ties or some other definitions. It is also possible that the winner set is empty, this is the case when there is no winner according to the election rule.

With regards to terminology, the terms "voting rule," "voting procedure," "voting scheme," "voting system," and, "voting protocol" are usually used as synonyms in the literature, but we draw the reader's attention to the preference of using the term "rule" or "function" when the emphasis is on the input-output relationship or the main aspect, or purpose of the rule being studied. On the other hand, the terms "procedure," "system," and "scheme," should be used when the emphasis is on the mechanism and steps taken to transform the input into an output. In addition, Conitzer [5] pointed out a similar note regarding the use of the term "voting protocol" in comparison with "voting rule." He mentions that the word "protocol" is used to further indicate "procedural aspects such as the manner in which the voters report their ranking (e.g., whether all voters submit their rankings at the same time or not)" [5].

1.3.2 Some Voting Rules

This section briefly lists a number of common and known voting rules. We will revisit some of these rules and study them in more detail in following chapters, where different computational aspects will be discussed in other contexts. Note that this list contains mostly the voting rules that will be discussed in later chapters (which includes most widely known and used rules). For an extensive list of voting methods, their classification, and properties of each, please see [10].

- **Majority**: the common simple rule by which a winner is a candidate who is chosen (ranked first) by a majority of voters. The term "simple majority" means"more than half of the voters." Very often, the term majority is used to stand for simple majority. It should be clear from the context whether a rule is a mojority or the simple majority

[3] There may be more than one winner.

rule. The simple majority rule is the unique rule that is anonymous, neutral, and strongly monotonic [4]. These terms will be defined later under *Properties of Voting Rules*.

- **Plurality**: the well-known rule by which a winner is a candidate who is ranked first by a plurality of voters. A plurality can simply be the majority or it can refer to some other group of voters that has the most influence on the outcome of the election (even if it is not the largest in number, i.e., the majority). A candidate wins by plurality if he gains more votes (by number or weight) than other candidates but the number of votes this candidate gets is not necessarily the simple majority of the votes.

- **Borda Count**: the rule (named after the French mathematician Jean-Charles de Borda) by which each candidate receives points according to his/her position in the voters ordering, where the candidate who is placed first gains more points than the candidate who is placed second and so on. The points are added and a candidate with the maximum number of points wins. Usually the top candidate in a vote is awarded $m - 1$ points, where m is the number of candidates, and the least preferred candidate gets 0 points.

- **Approval**: in approval voting, each voter specifies the candidates he/she approves, a candidate with the largest number of approvals wins. In approval voting, voters vote only once but a voter can vote for more than one candidate.

- **Veto**: the rule which is used to eliminate least preferred candidates. A candidate with the fewest number of vetoes wins[4].

The previous voting rules belong to a family of voting rules called scoring rules or positional scoring rules. In all of these rules, a vote is represented by a vector of integers $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$, where α_1 is the number points that goes to the candidate placed first in the vote, α_2 is the number of points that goes to the candidate placed second and so on. For example, in the Borda rule, $\alpha_1 = m - 1$ and $\alpha_m = 0$.

Ties can occur for different voting rules (at various stages of the rule for multistage rules). There are various ways for handling the issue of ties in elections. Adding another round of the same rule to determine the winner—if more than one candidate perform equally—is one example of tie-breaking technique. In fact, different tie-breaking rules correspond to distinct voting rules. Tie breaking can significantly affect the outcome of the election and also the computational complexity of the voting rule. In the following chapters, we will refer to the tie-breaking rule used when applicable. For most of the works covered in the survey, authors specify whether the voting method is a single-winner or multiple-winner method and what tie-breaking rule is used if a unique winner is to be elected. In some cases, however, a voting rule is studied in a general perspective without specifying what tie

[4]The veto method can also be thought of as similar to the so called "antiplurality" or "negative plurality" method. Since in antiplurality, a voter essentially votes *for all* but one candidate, i.e., the points in the vote/antiplurality go *against* the least preferred candidate. The use of either term will follow from the context.

handling procedures is used (in effect, ignoring the issue of ties or assuming no ties for the purpose of the analysis).We continue with more voting rules:

- **Plurality with run-off**: as its name suggests, this rule is similar to the plurality rule except that it proceeds in two rounds. In the first round, all candidates except the top two—according to their plurality scores—are eliminated and the votes favoring the eliminated candidates are transferred to the top two candidates as specified in the votes. A second round, the runoff, determines the winner.

- **Single transferable vote**: this rule consists of $m-1$ rounds, where m (as used throughout this chapter), is the number of candidates. In each round, the candidate with the lowest plurality score is eliminated and the votes for this candidate are transferred to the remaining candidates as specified in the vote. Note that a vote is a *complete* ordering over the candidates, so even after excluding some candidates, the votes favoring those candidates can still be used to rank the rest of the candidates.

- **Condorcet**: the Condorcet rule or Condorcet method is named after the Marquis de Condorcet, a French mathematician and philosopher. In this method, published in his "Essay on the Application of Analysis to the Probability of Majority Decisions" in French [17], the winner is the candidate who beats all other candidates in pairwise elections held to compare one pair of candidates at a time. For example, consider the election (C, V) where $C = \{a, b, c\}$ and $V = \{1, 2, 3, 4\}$, (here, V is explicitly the set of voters), and the voters vote as follows: $voter_1 : a > b > c, voter_2 : a > c > b$, $voter_3 : b > a > c, voter_4 : c > a > b$. Consider candidates a and b, a is preferred to b by three voters so a beats b. Now consider candidates a and c, a is preferred to c by three voters so a beats c. So a beats other candidates by a strict majority of votes (3 out of 4) so a wins this election by the Condorcet method. Unfortunately, this rule does not guarantee finding a winner. Consider the following example where $C = \{a, b, c\}$, $V = \{1, 2, 3\}$ and the preferences of the voters are as follows: $a > b > c$, $b > c > a$, and $c > a > b$. Every candidate beats one of the other candidates by a majority (two thirds) of votes. This example illustrates the so called Condorcet paradox (although this is not technically a paradox, the word "paradox" is used here to describe the counterintuitive observation that individual choices were rational, the rule is simple, yet the output (the social choice) is not rational) which was mentioned earlier. There are many other interesting observations related to the Condorcet method, reference [14] addresses some of these.

- **Sequential Pairwise Voting**: this rule proceeds as a sequence of head-to-head competitions. The order is set in advance, this is also referred to as the *agenda*. If a candidate wins the current competition, that candidate goes to face the next candidate on the preordered list. For example if the agenda is: a, b, c, (where a, b, and c comprise the candidate set) then a and b compete with each other first, then the winner competes with c. If two candidates tie at one stage, then each competes with the next candidate. The winner of the last pairwise comparison is the winner of the overall election.

- **Cup (or Binary Voting)**: this rule is represented using a balanced binary tree. Leaf nodes of this binary tree represent candidates (who are assigned to leaf nodes using some schedule or using randomization, in the case of a randomized candidate-assignment to leaf-nodes the rule is called Randomized Cup). The parent of two leaf-nodes is the winner of the pairwise election of the two leaf nodes. The assignment of winners to non-leaf nodes proceeds this way until we reach the root node which is the winner of the election.

All of the rules listed above are typically unique-winner methods (except for Condorcet's rule which by definition *always* elects a single winner if one exists or no winner if the Condorcet winner does not exist). The next four rules are multi-winner rules. Furthermore, these four rules are k-winner rules. The number of possible winners k is fixed before an actual election takes place:

- **k-Approval**: A special case of approval voting is the k-approval rule in which the k candidates with the k largest numbers of approvals win.

- **Bloc (a.k.a. block)**: in bloc voting, each voter gives one point to each one of k candidates that the voter wants to elect as winners. The k candidates with the most points win.

- **Cumulative**: in cumulative voting, each voter distributes a fixed number of points among the k candidates. The k candidates who gain most points win. The distribution of points among candidates can be thought of as a way to express the intensity of a vote, since in addition to voting for the favorite k candidates, points are distributed among these k candidates such that the *most* preferred gets most points and the least preferred, although among the preferred candidates, gets the smallest number of points. It is argued that cumulative voting is especially interesting because it gives minorities a better chance for representing their preferences [4] .

- **Single Non-Transferable Vote (SNTV)**: in this voting rule, each voter gives one point to a favorite candidate, the k candidates with most points win the election.

For the following rules we need to define $N(x, y)$. Let x and y be two candidates and let $N(x, y)$ be the number of votes preferring x to y.

- **Maximin (a.k.a. Simpson)**: the maximin score of a candidate x is denoted by $s(x)$ and is given by $s(x) = min_{y \neq x} N(x, y)$, that is, x's maximin score is the lowest score it gets in any pairwise election where a pairwise election score is equal to the number of voters who prefer x over the opponent. The larger the maximin score, the higher the candidate is ranked.

- **Copeland (a.k.a. Tournament)**: the Copeland score of a candidate is the number of pairwise elections that the candidate wins minus the number of pairwise elections the candidate loses[5] [20, 21] . The global election proceeds through pairwise elections for each pair of candidates in turn. A candidate gets 1 point when defeating

[5] The Copeland score of a candidate is the sum of the pairwise elections that the candidate wins minus the sum of the pairwise elections he/she loses, or simply it is the sum of the pairwise elections that the candidate wins. If two candidates tie in a pairwise competition, then neither gets a point.

an opponent in a pairwise election, -1 point when losing and 0 points in case of a tie. Alternatively, the Copeland rule can be described using $N(x, y)$ as follows: for any two candidates x and y, let $C(x, y) = 1$ if $N(x, y) > N(y, x)$, $C(x, y) = 1/2$ if $N(x, y) = N(y, x)$, and $C(x, y) = 0$ if $N(x, y) < N(y, x)$. The Copeland score of a candidate $s(a) = \Sigma_{y \neq x} C(x, y)$. In either cases, candidates are ranked by their scores; the higher the score, the higher the candidate is ranked.

- **Llull's** rule is defined similarly as Copeland rule except for tie-breaking. In this rule, proposed by Ramon Llull—a Catalan writer and philosopher from the thirteenth century—if two candidates tie then *each* one of them gets a point [11] .

- **Bucklin**[6]**(a.k.a. Grand Junction system**[7]): In this rule, if a candidate uniquely gets a majority of first choice votes, then that candidate is the winner. Otherwise, if a candidate has a majority of first and second choice votes, that candidate is the winner. Otherwise, if a candidate has a majority of first, second, and third choice votes, then that candidate is the winner, and so on. If two candidates get a majority at the same stage of the rule, then the candidate with the larger total at that stage is the winner. Alternatively, for any candidate i and integer l, consider the number of voters that rank i among the top l candidates, if a voter ranks l candidates as top candidates, then l is increased repeatedly until some candidate is ranked among the top candidates by a majority of voters.

A Condorcet-consistent rule is a voting rule electing the Condorcet winner whenever there is one, however it finds winners that are closest to being Condorcet winners if the Condorcet method does not output one. Closeness is measured in terms of the distance between the current voters profiles and the profiles that yield a Condorcet winner under the condition of making minimal changes. These changes try to remove the "inconsistencies" (induced by conflicting preferences) in the votes so that a winner can be found. The following rules are Condorcet-consistent:

- **Dodgson's** rule: a voting rule proposed in 1876 by the mathematician and writer Charles Lutwidge Dodgson (Lewis Carroll) which reemerged in [2] and summarized later in [20] . The election is won by the candidate(s) who is (are) closest to being a Condorcet winner where each candidate is given a score that is equal to the smallest number of exchanges of adjacent preferences in the voters' preference orders that are needed to make the candidate a Condorcet winner (with respect to the resulting collective preference profile). The candidate (or candidates in the case of a tie) with the lowest score is (are) the winner(s).

- **Kemeny's** rule: a voting rule proposed by Kemeny [15, 16]. Kemeny's rule is a preference aggregation rule that consists of aggregating n individual preference profiles into a collective profile (called Kemeny consensus) being closest to the n profiles, with

[6] A formal definition of Bucklin rule appeared in [6] and another definition appeared in a survey of voting methods available at:
`http://fc.antioch.edu/~james_green-armytage/vm/survey.htm`

[7] Named after Grand Junction, (Colorado,) the city where the Bucklin rule was first proposed.

respect to a distance which is the sum, for all voters, of the numbers of pairs of alternatives on which the social profile disagrees with the voter's profile. A Kemeny-rule winner is a candidate who is ranked first in one of the Kemeny consensus. Kemeny's voting rule was shown to be the unique rule that is neutral, consistent, and Condorcet consistent [30]. These properties are defined in the following section.

- **Young's** rule: a voting rule introduced by Young in 1977 as an extension to the Condorcet rule [31]. A winner is a candidate who becomes a Condorcet winner after removing the least number of votes, these votes can be thought of as the votes which cause the inconsistencies or ties that prevent having a Condorcet winner (compare this to the following observation about Plurality: after removing a number of votes, a winner is the candidate who is placed first in all of the remaining votes).

- **Black's** rule: suggested by Black in 1958 [2] to recognize the importance of the Condorcet criterion yet exploit the advantages of the Borda count. Black's rule elects the Condorcet winner if there is one, otherwise the winner under this rule is the candidate with highest Borda count.

Another rule that attempted to avoid the Condorcet paradox is the following:

- **Slater** rule: a preference aggregation rule that aggregates n individual profiles into a collective profile (called Slater ranking) such that the number of pairs of candidates on which the final ranking disagrees with the majority vote is minimized. Of course, more than one Slater ranking is possible.

1.3.3 Properties of Voting Rules: Criteria for Evaluating Voting Rules

The following list contains some properties that are considered when evaluating a voting rule:

- **Anonymity** embodies the notion of treating voters equally. A voting rule is anonymous if the outcome of the election remains the same under any permutation of the votes/voters. So if voter 1 and voter 2 switch their votes then the outcome is the same, what really matters is the vote (and its weight), *who* casts the vote is not a factor in determining the outcome. Therefrom is the link between anonymity and the treating voters equally.

- **Neutrality** captures the notion of treating *candidates* equally. For example, if the preferences expressed in the votes remain the same[8] and candidates a and b switch their names, then the outcome of a *neutral* voting rule—under this change—will be identical to the outcome of the original election except with positions a and b also switched. This shows that the relative ranking of a and b is unaffected by their names and is only determined by the preferences of voters.

[8] Of course for the preferences to be the same this change is also expressed in the votes by replacing every a with b and every b with a in the original votes.

- **Monotonicity (a.k.a. Positive Response)** stands for the property of maintaining a candidate's position if he/she gains more support from voters, under the condition that the relative order of other candidates remain the same. If gaining more support from voters may cause the candidate to be ranked lower or to lose under a voting rule then that voting rule is not monotonic. Strong monotonicity implies that ranking a candidate higher in the votes (with relative order of other candidates unchanged), must result in ranking the candidate higher in the social ranking.

- **Strong Monotonicity** implies monotonicity with the addition that gaining more support (strictly) improves the position of a candidate.

- **Pareto Efficiency** resembles to the principle of *unanimity* when all voters agree on the relative ranking of some candidates. For example, if all voters prefer candidate a to candidate b then the outcome of a *Pareto-efficient* voting rule also prefers a to b. This is also linked to the term *Pareto-optimality* and the condition that if a candidate is ranked lower than some other candidate by all voters, then that candidate never wins under a Pareto-efficient (or Pareto-optimal) voting rule. This property is named after Italian welfare economist Wilfredo Pareto.

- **Independence of Irrelevant Alternatives (IIA)** has already been defined as a condition of a social preference function in the previous mention of Arrow's theorem. For voting rules, this criterion can be rephrased to mean whether a voting rule outputs the same result if non-winning (irrelevant) candidates are added or removed.

- **Non-dictatorship** which is a universally agreed-upon condition for acceptable voting rules. This condition says that no voter alone can determine the outcome of the election regardless of the preferences of other voters. Non-dictatorship is typically an implied condition for all voting rules.

There are other properties/criteria used to evaluate voting rules such as:

Condorcet-Consistency (or Condorcet Criterion) which is another property of voting systems (though not necessarily viewed as an applicable criterion when considering all voting systems). A Condorcet-consistent rule is one that declares as winner the Condorcet winner if there exists one. Since the Condorcet-method is based on binary (pairwise) comparisons, the Condorcet criterion may not apply to voting systems where the ranking is not merely based on pairwise comparisons.

Extended Condorcet Criterion (ECC) was proposed by Truchon in 1998, this criterion extends the Condorcet criterion to a partition of the set of alternatives into two subsets C_1 and C_2, such that for any $x \in C_1$ and $y \in C_2$, if the majority prefers x to y, then x must be ranked above y in the final ranking [28]. This criterion extends the Condorcet criterion in the sense that the latter ranks an *individual* alternative x above y if the majority of voters prefers x to y, and in the extended Condorcet criterion, if the set of alternatives can be partitioned such that all alternatives in a *subset* C_1 of the partition are preferred to all alternatives in other *subsets* then C_1, as a subset of alternatives, should be ranked higher than other subsets.

Consistency if the set of voters is split into two disjoint sets and each set votes separately and elects the same candidate(s) as winner(s) then the voting rule is consistent if the result of the overall election, where voters of the two subsets vote together as a single set of voters, declares the same winner(s) as the two disjoint elections. Formally, this is denoted as $\mathcal{E}(V_1, C) \cap \mathcal{E}(V_2, C) \subseteq \mathcal{E}(V, C)$ where (V_1, V_2) is a partition of the voter set V.

Strong Consistency as the name suggests is a strong form of consistency. A voting rule is strongly consistent if for any partition of the voter set V into V_1 and V_2 then $\mathcal{E}(V_1, C) \cup \mathcal{E}(V_2, C) = \mathcal{E}(V, C)$. In other words, the winner of the overall election is independent of the "subelection" in which he/she wins.

Weak Axiom of Revealed Preference which requires that if a candidate c is a winner of some election then that candidate is also a winner of any election where the candidate set is a *subset*—containing c of course—of the original set [24].

Path Independence is satisfied by a voting system if for any voter set V and candidate set C, the outcome of the election $\mathcal{E}(V, C)$ is the same as the outcome of $\mathcal{E}(V, \mathcal{E}(V, C_1) \cup C_2)$, where C_1, C_2 form a partition of C [23, 24].

1.4 Comments and Bibliographic Notes

- A related topic that has not been addressed here is the topic of ower indices. For more on the theory and mathematics of power indices, see [26]. We note the relative sparseness of studies on this subject, in comparison to the multitude on voting. However, the reader is referred to [13, 18, 19, 29, 12] as examples of progress in this direction.

- In this chapter, we have briefly discussed Arrow's theorem and the phenomenon of cyclic majorities that are both considered among the paradoxes of voting. More on voting paradoxes can be found in [7, 22, 27].

- For more on the axiomatic foundation of ranking and the mathematics of voting, see [16] and [8]. For rigorous mathematical definitions of voting rules and their properties see [4] and for a summary analysis of voting procedures see [21].

Chapter 1 Bibliography

[1] Arrow, K. *Social Choice and Justice*, vol. 1 of *Collected Papers of Kenneth J. Arrow*. The Belknap Press of Harvard University Press, 1983.

[2] Black, D. *The Theory of Committees and Elections*. Cambridge University Press, Cambridge, 1958.

[3] Blau, J. A direct proof of Arrow's theorem. *Econometrica 40*, 1 (1972), 61–67.

[4] Brams, S., and Fishburn, P. Voting procedures. In *Handbook of Social Choice and Welfare*, K. J. Arrow, A. K. Sen, and K. Suzumura, Eds., vol. 1 of *Handbook of Social Choice and Welfare*. Elsevier, 2002, ch. 4, pp. 173–236.

[5] Conitzer, V. *Computational Aspects of Preference Aggregation*. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA, 2006.

[6] Conitzer, V., and Sandholm, T. Communication complexity of common voting rules. In *EC ' 05: Proceedings of the 6th ACM Conference on Electronic Commerce* (New York, NY, USA, 2005), ACM Press, pp. 78–87.

[7] Fishburn, P. Paradoxes of voting. *The American Political Science Review 68*, 2 (1974), 537–546.

[8] Fishburn, P. Discrete mathematics in voting and group choice. *SIAM Journal on Algebraic and Discrete Methods 5*, 2 (1984), 263–275.

[9] Gaertner, W. *A Primer in Social Choice Theory*, first ed. Oxford University Press, Oxford University Press Inc., 2006.

[10] Green-Armytage, J. A survey of basic voting methods. `http://fc.antioch.edu/~james_green-armytage/vm/survey.htm`.

[11] Hagele, G., and Pukelsheim, F. The electoral writings of Ramon Llull. *Studia Lulliana 41*, 97 (2001), 3–38.

[12] Hemaspaandra, E., and Hemaspaandra, L. Computational politics: Electoral systems. In *MFCS '00: Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science* (London, UK, 2000), pp. 64–83.

[13] Hemaspaandra, L., Rajasethupathy, K., Sethupathy, P., and Zimand, M. Power balance and Congressional apportionment algorithms. Technical report, Computer Science Department, University of Rochester, Rochester, NY, USA, 1996.

[14] Johnson, P. *Social Choice: Theory and Research.* No. 123 in Quantitative Applications in the Social Sciences. Sage Publications, Inc., Thousand Oaks, CA: Sage, 1998.

[15] Kemeny, J. Mathematics without numbers. *Daedalus 88* (1959), 575–591.

[16] Kemeny, J., and Snell, J. *Mathematical Models in the Social Sciences.* Introduction to Higher Mathematics. The MIT Press, 1973. Second MIT press edition.

[17] Marquis de Condorcet. Essai sur l'application de l'analyse à la probabilité des decisions rendues à la pluralite des voix, 1785.

[18] Matsui, T., and Matsui, Y. A survey of algorithms for calculating power indices of weighted majority games. *Journal of the Operations Research Society of Japan 43*, 1 (2000), 71–86.

[19] Matsui, Y., and Matsui, T. NP-completeness for calculating power indices of weighted majority games. *Theoretical Computer Science 263*, 1,2 (2001), 305–310.

[20] Niemi, R., and Riker, W. The choice of voting systems. *Scientific American 234* (1976), 21–27.

[21] Nurmi, H. Voting procedures: A summary analysis. *British Journal of Political Science 13*, 2 (1983), 181–208.

[22] Nurmi, H. Voting paradoxes and how to deal with them. *Public Choice 107*, 1,2 (2001), 194–197.

[23] Plott, C. Path independence, rationality, and social choice. *Econometrica 41*, 6 (1973), 1075–1091.

[24] Plott, C. Axiomatic social choice theory: An overview and interpretation. *American Journal of Political Science 20*, 3 (1976), 511–596.

[25] Roberts, F. *Group Decisionmaking.* Discrete Mathematical Models with Applications to Social, Biological, and Environmental Problems. Prentice-Hall, Englewood Cliffs, NJ, USA, 1976, ch. 7, pp. 425–472.

[26] Straffin, Jr, P. *Topics in the theory of voting.* The UMAP expository monograph series. Birkhauser Verlag AG, 1980.

[27] Tabarrok, A. Paradoxes of voting. Available at `http://mason.gmu.edu/~atabarro/paradoxofvoting.pdf`.

[28] Truchon, M. An extension of the Concordet criterion and Kemeny orders. Cahiers de recherche 9813, Université Laval, Département d'économique, 1998. Available at `http://ideas.repec.org/p/lvl/laeccr/9813.html`.

[29] Uno, T. Efficient computation of power indices for weighted majority games. Technical Report NII-2003-006E, National Institute of Informatics, Japan, 2003.

[30] Young, H., and Levenglick, A. A consistent extension of Condorcet's election principle. *SIAM Journal on Applied Mathematics 35*, 2 (1978), 285–300.

[31] Young, H. P. Extending Condorcet's rule. *Journal of Economic Theory 16*, 2 (1977), 335–353.

Chapter 2

Classification of Related Work

2.0 Prologue

The classification of work related to the computational aspects of voting can be approached in different ways. An approach typically depends on the perspective, the discipline the classification is based on, and the broadness of the coverage.

Since this is the first survey of the topic *computational aspects of voting systems* as such, and since classification of related work is one of the main contributions of a literature survey, no classification of topics in computational aspects of voting systems, as a field, has been proposed yet.

This chapter provides a list of possible classifications, and concludes with the classification adopted in the organization of this survey.

2.1 Mathematics-based Classification

A possible classification is one that is based on the mathematical tools, theories and techniques used in a work. There are basically two different models discussed in social choice theory: discrete models and spatial models.

Problems in discrete models are tackled using techniques from discrete mathematics and logic, whereas spatial models draw heavily from calculus of continuous sets, real analysis and geometry. In addition to these two approaches, statistical methods are also used to study various issues in voting and social choice.

Most of voting aspects discussed in this thesis fall in the category of discrete models. Welfare economics topics, on the other hand, fall mostly in the category of spatial continuous models. Therefore, this mathematics-based classification is suitable for an overall classification of *social choice* theory and applications, which encompasses other topics besides voting.

This classification was adopted in a work presenting social choice theory and research by Johnson [2]. Although the classification was not explicitly set forth, it was reflected in the sectioning of this broad albeit introductory brief paper.

2.2 Direction

Computational aspects of voting systems comprise a topic that clearly combines both computer science and social choice theory, in particular via voting.

Related work has two main directions: The first direction goes from computer science to social choice theory. Work in this direction applies knowledge from computer science to solve classical problems in social choice theory. Mostly, Artificial Intelligence- and Complexity Theory-based analyses of voting systems is found in this direction.

The second direction goes the other way, from social choice theory to computing. Work in this direction imports concepts and techniques from social choice theory to solve problems originally arising in computer science.

The next two chapters of this thesis are dedicated to surveying work in the first direction while the last two chapters present work going *mainly* in the second direction except for some parts of the last chapter (on electronic voting) that combines the two directions in balance.

2.3 Analysis versus Design

Studies on voting systems can also be categorized into studies on existing systems and studies on designing new systems.

The first category includes studies on properties of voting systems, votes and preferences and how to represent them, computational complexity of existing voting systems and related problems, impossibility and impracticality results, approximation and heuristic algorithms for solving problems pertinent to known voting schemes, dichotomy results, applications of known voting rules in different computing-related disciplines, etc.

The second category includes topics as broad as mechanism design and axiomitization of voting procedures and related topics, and as specific as dichotomy criteria, meta-characterization of voting systems and proposals for designing reliable voting schemes.

The classes according to this classification, however, can be argued to be not clearly distinct. The two categories intersect at some topics, for example dichotomy results derived to apply to existing systems can also be used as criteria for designing new systems. Also, proposals for designing new systems with desirable properties are sometimes mere modifications of existing ones. Even the very impossibility and impracticality results derived from observations of currently used systems can be thought of as a step towards formalizing criteria for designing new acceptable systems.

2.4 MCDM vs. DMU vs. Social Choice

Multiple Criteria Decision Making (MCDM), Decision Making under Uncertainty (DMU), and Social Choice (SC) can be viewed as three broad classes of topics within Decision Mak-

ing. This classification was brought to the author's attention while reading a paper "On the limitations of Ordinal Approaches to Decision-making" [1].
Although this classification reasonably divides problems in decision making, individual work often combines more than one of these headings. It is very common to find work discussing social choice issues that is also considered a work in MCDM.

Much of the work surveyed here is pertinent to both social choice and multiple criteria decision making.

2.5 *This* Classification

The classification proposed here is based on the *computer science* content of the surveyed work. A nice feature of this classification is that it is orthogonal to subjects in computing. The following chapter, for example, discusses aspects related mainly to Artificial Intelligence, the chapter after that discusses the complexity-theoretic aspects of voting systems which is—as the title suggests—related to Algorithms and Complexity Theory, the remaining chapters overview applications of voting in computing, the applications are classified based on the subject they are applied in for example: Networking, Databases, Cryptography and Security, etc.

Figure 1 depicts the taxonomy proposed here and reflected in the chapter-theme organization of this thesis[1].

[1] No chapter is dedicated entirely to discussing design aspects of voting systems, but related design aspects are discussed where relevant in different chapters.

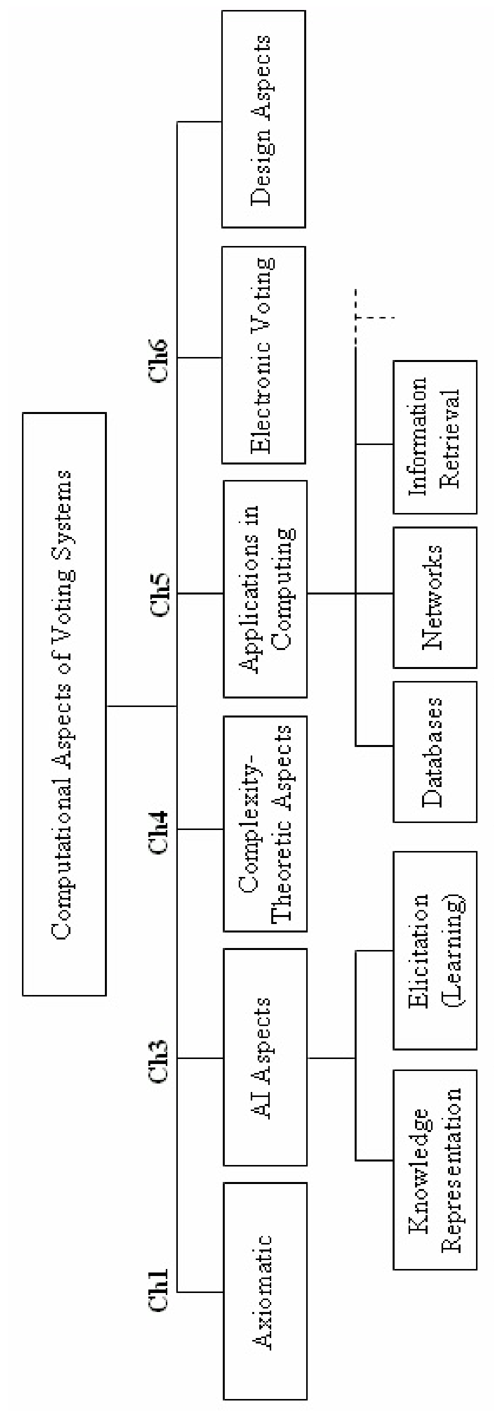

Figure 2.1: A proposed taxonomy of computational aspects of voting systems.

Chapter 2 Bibliography

[1] Dubois, D., Fargier, H., and Perny, P. On the limitations of ordinal approaches to decision-making. In *KR' 02: Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning* (Toulouse, France, 2002), pp. 133–146.

[2] Johnson, P. *Social Choice: Theory and Research*. No. 123 in Quantitative Applications in the Social Sciences. Sage Publications, Inc, Thousand Oaks, CA: Sage, 1998.

Chapter 3

Preferences: Representation and Elicitation

The core of a (non-collegial) voting game on a compact policy space W is non-empty only for a nowhere dense set of preferences, where preferences are continuous and endowed with the closed convergence or C^0-topology: that is emptiness of the voting core is generic in this topology (Le Breton, 1987). On the other hand, if preferences are continuous and convex and that dimension of the space is suitably bounded then a core does exist.

Norman Schofield (from the introduction of a paper on preferences)[1]

3.0 Prologue

Input specification is essential in the processing of any computable entity. This also applies to social choice functions and voting systems. In the previous chapter we have described a number of voting systems and listed properties of voting rules. In the next chapter, we will discuss the computational complexity of various problems related to voting systems. This will be in light of some general assumptions about the input to the voting system, which is constituted by the set of candidates or alternatives and the set of voters and their preferences.

Before we explore the computational aspects of input representation—which amounts to preferences in this domain—let us revisit a definition stated in Chapter 1 as we postponed the discussion of details to this chapter.

In Chapter 1, we have defined an election to consist of a finite set of alternatives and a finite set of agents who vote over the set of alternatives. For simplicity, the assumption was that both the candidates and voters are represented as sets and these sets are explicitly and entirely encoded as the input to an election rule. Also, we have assumed that the votes cast are complete and explicit orderings over the set of alternatives and that every vote lists all

[1]Schofield, N. (1994) "The C^1 topology on the space of preference profiles and the existence of a continuous preference aggregation." School of Business and Center in Political Economy, Washington University in St. Louis, unpublished.

the alternatives in order of preference.

These assumptions, although plausible and suitable for an introductory chapter, are not accurate enough for a technical discussion of topics in preference aggregation. Before we discuss the various aspects of preference representation in voting settings, we first have to revisit our definition of elections.

In Chapter 1, an election consists of two sets, the set of candidates and the set of voters. This definition captures the main constituents of an election but it is not precise mathematically speaking. The following is a revision of that definition:

Definition 3.0.1 (Election Definition (revision))**.** *An election consists of a* list *of alternatives called candidates C, and a* multiset *of votes representing voters. The vote of the ith voter is expressed by a* list $\pi_i \in \Pi(C)$ *where* $\Pi(C)$ *is the set of all permutations of the candidate set.*

Feasible candidates are expressed by a list since a list is an ordered set and the order is introduced to ease referring to candidates (especially when names are ignored) by using their position in the list as an index. This order is independent of any preference or any position the candidates may have on a particular issue; the order is merely introduced to facilitate the process of identifying and referring to candidates.

The set of voters can be represented by their votes since voting rules ought to be anonymous and only the vote not *who* casts the vote should be considered in the final outcome. The votes are input as a multiset since more than one voter may have the same vote which results in multiple entries in the vote set. This is more likely to happen when the set of voters is much larger than the set of alternatives. In this case, every vote can be associated with a number that refers to the multiplicity of the vote, in other words the number of voters whose preferences are represented by this particular vote. The set of votes can also be ordered to facilitate referring to votes (or voters) although this is less needed in comparison with referring to candidates, for example when casting the votes or presenting candidates and discussing preferences.

According to this revision, a voting rule is a mapping from the *multiset* of votes that selects a winner $c \in C$ (the range is possibly a set of winners or an ordering of the alternatives depending on the function of the voting rule in a particular setting). Having refined the definition of a central entity in the discussions of this thesis, we next move to the representation of the input, i.e., the preferences of agents or voters over a set of alternatives or candidates.

Since preference representation is essential to any preference aggregation method, much of the content of this chapter applies to **(combinatorial) auctions** and **exchange (negotiation)** as well as **voting** since all share the essence of being preference aggregation methods.

The next section describes three "visual" representations of a given preference aggregation setting. The objective is to introduce preference representation using a vivid discussion that shows the need for and purpose of having different approaches for representing preferences, which in turn explains the richness of the literature on preference representation and preference elicitation. The following section on *Preference Representation* discusses the motivation behind preference representation in various domains, gives examples of some preference representation languages and explains some important properties of preferences, such as single-peakedness. The section on *Preference Elicitation* defines preference elicitation with respect to different preference aggregation domains, and lists some results and applications. As in the previous chapters, this chapter is appended with *Comments and Bibliographic Notes*.

The reason behind discussing preference representation and preference elicitation in the same chapter is that these two topics are closely related; preference elicitation methods usually take into consideration the representation of preferences to be elicited. Although this chapter does not go into detail on either topic in relation to the other, it touches on the basic concepts in each, emphasizes the related computational aspects in the voting context and presents related work and results. Some of the results are pertinent to the computational complexity of problems in preference representation and preference elicitation. Although these results are mentioned in passing here and some will be revisited in the next chapter, the reader may skip to the first section of the next chapter for a brief introduction to complexity theory before proceeding here.

3.1 Three Pictures Depicting Preferences

3.1.1 Preference Graphs

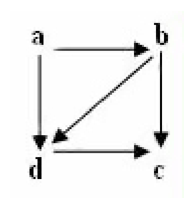

Figure 3.1: An example of a pairwise election graph.

An election graph is a directed graph where the vertices represent candidates and an edge from vertex a to b represents a vote that prefers a to b. The weight on the edge from a to b (if present) represents the number of voters preferring a to b.

The same election graph can represent the final outcome of the election rule. In this case the graph is not weighted and each edge stands for the preference between two candidates in the final (social) preference.

A cycle in an election graph (the graph of the result of the election) represents a cyclic preference in the votes when aggregated. For example, the following directed edges (a, b), (b, c), (c, a) in a graph with three vertices a, b, and c, visually depicts the simple example of the famous Condorcet paradox when three voters have the following preferences: $a > b > c$, $b > c > a$, and $c > a > b$. A similar cycle in the graph representing the actual preferences of the voters with the weights of the three edges being all equal depicts the same case.

A majority graph $M(P)$ is defined similarly. The set of vertices is the set of candidates C and for all $a, b \in C$, there is a directed edge from a to b if and only if the number of voters who prefer a to b is greater then the number of voters who prefer b to a. The majority graph is directed and no edge is bidirectional (i.e., $M(P)$ is asymmetric) and contains no loops (i.e., $M(P)$ is irreflexive). If the number of voters is odd, the majority graph is also complete, that is for each $a, b \in C$, either an edge directed from a to b exists or an edge directed from b to a exists in the graph. A graph $G = (V, E)$ that is complete, asymmetric and irreflexive is also called a *tournament* on V [26]. Therefore, when the number of voters is odd the majority graph $M(P)$ is a tournament on C.

Graph representation of elections is sometimes used to tackle problems using techniques and algorithms from Graph Theory.

3.1.2 Preference Curves

A preference curve is another visualization of preferences in a preference aggregation setting. The figure below shows the preferences of an agent and to the right shows the preference curve of that agent. The horizontal axis represents alternatives and the vertical axis represents the order[2] of the preference. The picture is self-explanatory, the absolute maximum which is the alternative on the horizontal axis corresponding to the largest point on the vertical axis is the most preferred alternative and also the winner under the majority rule. The agent is indifferent between two alternatives that correspond to the same point on the vertical axis, and the alternative corresponding to the lowest point on the vertical axis is the least preferred alternative.

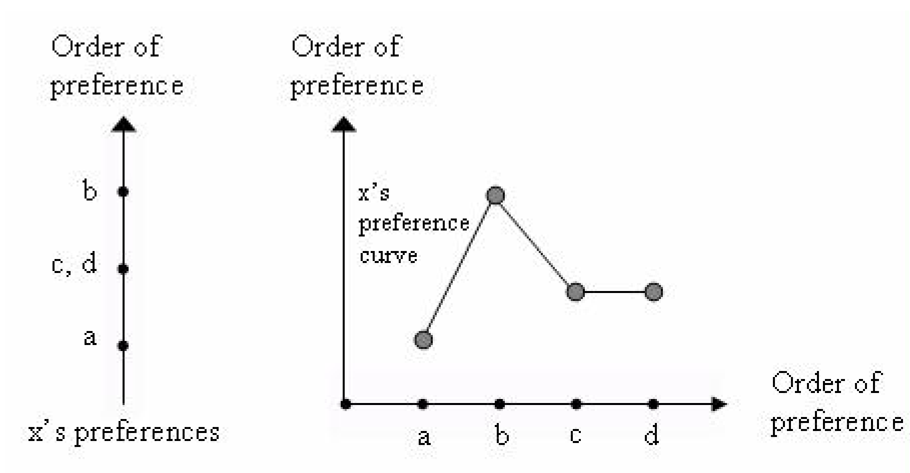

Figure 3.2: An example of a preference curve.

[2]But not necessarily the intensity of the preference.

The absolute height of the points on the vertical axis and the exact slope of the lines connecting the different points on the curve have no significance unless stated otherwise. An example of that special case would be a preference curve depicting preferences in a quantitative preference structure. Also, the continuity and behavior of the lines connecting the points are irrelevant unless the model is a continuous model and the curve is explicitly stated to represent things like the intensity and intra-comparability of preferences.

Figure 3.3 has a number of preference-curves, each corresponding to an agent in a multiagent system or to a panel of legislators voting over a number of policies put forward. Not only does this picture allow studying the preference of each individual in the group but it also allows inter-comparability between preferences and ultimately assists the preference aggregation process.

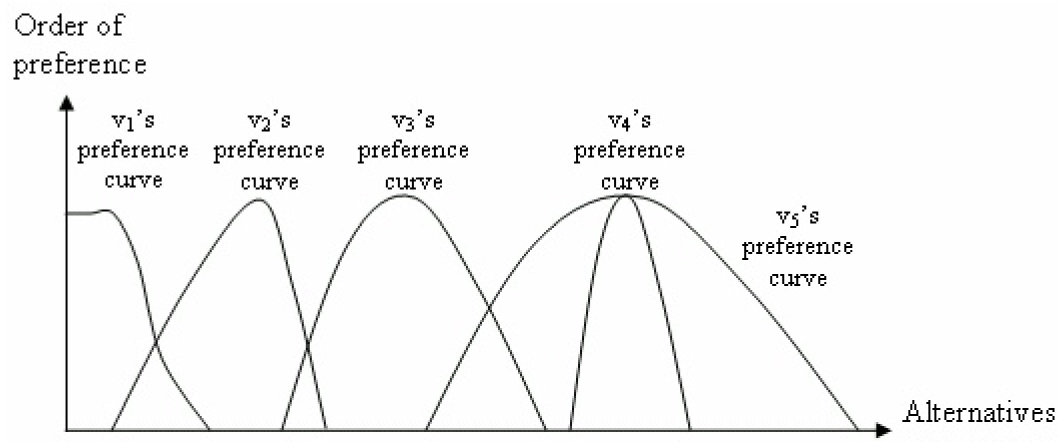

Figure 3.3: An example of a collection of preference curves.

3.1.3 Preference Matrices

Although a matrix is not typically perceived as a visual structure, many visual observations when looking at matrices directly lead to some profound conclusions. The use of matrix notation in preference representation was first seen in a paper by Black [3], in which he acknowledges Newing's suggestion of using matrix notation. Let us now explore some properties of the matrix construct in representation and aggregation of preferences.

The preference matrix A is a square matrix $m \times m$, where m is the number of alternatives. The diagonal of the matrix—across entries indexed by the same row and column number, (i, i), where $i \in \{1, \ldots, m\}$—is drawn before inserting the value of any other entry. For any other entry (i, j) where $i, j \in \{1, \ldots, m\}$, the entry is an ordered pair calculated as follows: the alternatives a_i and a_j are compared, the first element in the ordered pair M_{ij} is equal to the number of agents preferring a_i to a_j and the second element in the ordered pair M_{ij} is equal to the number of agents preferring a_j to a_i. Note that only the upper right triangular portion of the matrix or the lower left triangular portion of the matrix needs to be calculated. The other half of the matrix can be obtained from the one computed initially by a simple symmetry observation. If n voters prefer a_i to a_j and p voters prefer a_j to a_i, where $n, p < m$ and $n + p \leq m$ and this is recorded as $M_{ij} = (n, p)$ then similarly and based on the same information about agents preferences we can write $M_{ji} = (p, n)$. Generally, each row to the right of the main diagonal is a reflection—about the diagonal—of the column immediately beneath it.

Although the construction of this matrix seems to be a tedious job, one can easily infer results regarding the pairwise majority rule by just looking down the columns of this matrix. The same result may require at least as much work by considering the preferences of individuals and yet is less obvious by merely looking at the preferences. Moreover, many other results that are related to pairwise comparisons between the alternatives can be easily arrived at using the matrix notation which is constructed only once (as a preprocessing phase) while some computations might need to be repeated every time a question need to be answered when using the individual orderings of the alternatives.

For example, the figure below shows the preference schedules of the agents to the left and to the right shows the preference matrix constructed as explained above. If you are asked to hide the matrix portion of the figure and find the pairwise majority rule winner given these preferences, you will need some time despite the small size of the problem, five agents and six alternatives. By looking at the preference matrix, one can immediately find the winner by looking down the third column (or across the third row) and noticing that a_3 is preferred to any other alternative by a majority of voters. From this figure too, one can quickly see that a_4 is preferred to a_6 by *all* agents.

So far, we have seen three different ways of visually representing preferences. There are, indeed, different ways and approaches for representing preferences, visually, logically and using mathematical notation and constructs. The specific choice depends on the purpose of the representation, the nature of the application domain, the scale, and the level of abstraction in the context. An example of the first dimension, the purpose or goal of the representation, can be given using the three visual representations described above. Graph epresentation of preferences or elections can be used to detect cycles corresponding to cyclic majorities or local cycles[3] in the preferences of the voters, it can also be used in the analysis of some voting rules such as the Kemeny and Slater rules as we will see in Chapter 3 of this thesis. Techniques from Graph Theory are broadly used to obtain results using graph representations. Preference curves are used to study intra- and inter- preference properties that can assist deducing results about the aggregation process and what sort of outcome to expect. Later in this chapter, we will study some of these properties in more detail. Matrix representation, although not as widely used as other representations, have led to some discoveries in preference-related problems such as the Stable Roommate problem. In this problem for example, the problem of finding a "stable" matching for $2n$ people to be-

[3]Cycles that are formed by the preferences of a strict subset of the agents. This type of cycle does not necessarily lead to the Condorcet paradox or to declaring no winner.

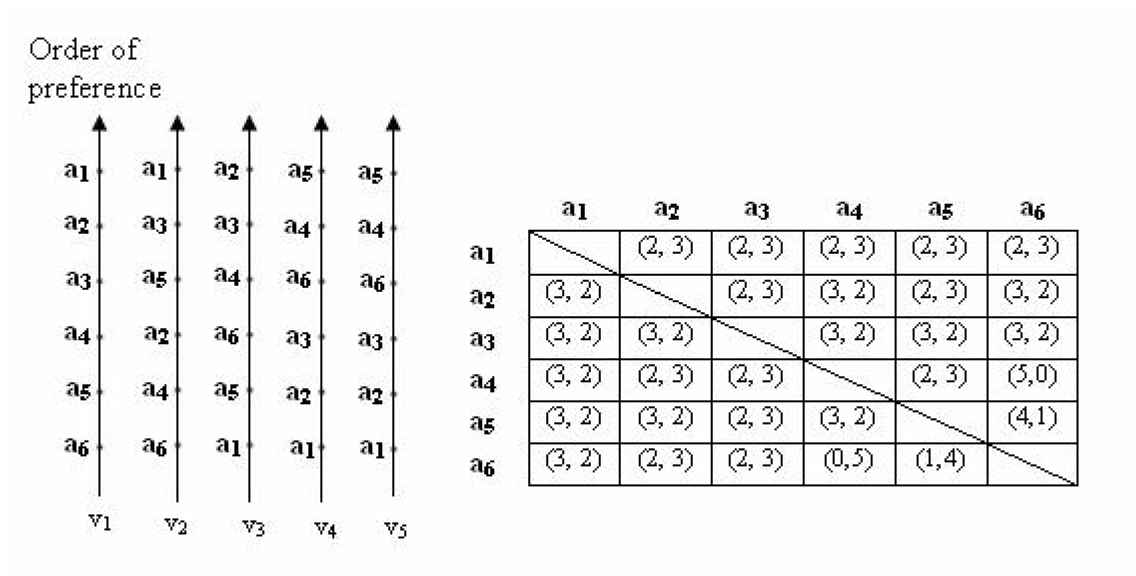

Figure 3.4: An example of a preference matrix.

come roommates is reduced to the problem of having consecutive 1's in rows of a matrix representing preferences.

We now move one step higher in the level of abstraction, and discuss preference representation in concept, motivation, and by way of example.

3.2 Preference Representation

3.2.1 Motivation

Voting is a general method for aggregating preferences in many settings. These include natural settings in which human voters express their preferences over a set of alternatives. Examples of these natural settings are presidential elections and job candidate interviews. The domain of voting also extends to multiagent systems in which software agents vote over alternatives that may be webpages or other software agents. The latter corresponds to ranking systems where the set of agents and the set of alternatives coincide. In many of these settings the space of voters and/or alternatives can be a combinatorial (very large) domain which makes the search space, to be explored for optimal result, grow prohibitively.

A combinatorial domain is defined as a Cartesian product of a finite value domain for each member of a set of variables, an alternative in a combinatorial domain is a tuple of values. As the number of values increases (linearly), the size of the corresponding combinatorial domain grows exponentially. In such settings, fully and explicitly expressing the preference of each agent over all alternatives may not be feasible. Therefore, the representation of preferences should be concise in order to spare time and space resources.

Different preference representation languages were proposed as a remedy to this problem. Also, some reference elicitation techniques are suggested such that only relevant *parts* of the vote are considered at a time and not the whole vote.

In addition to the peculiarities of combinatorial domains, preference structuring and preference representation languages (PRL) are also suggested as a way to prevent undesirable situations in voting such as impossibility results and paradoxes.

Following sections of this chapter discuss preference representation languages and preference elicitation in more depth.

3.2.2 The Domain

Any preference aggregation setting involves two basic entities: a set of alternatives or candidates and a set of agents or voters expressing their preferences over the set of alternatives. The preferences are structured in a specific agreed-upon way among the agents and a preference representation language encodes these preferences (1), is used to elicit preferences in case of partial preference description (2), and is used to aggregate the preferences into an outcome according to some defined rule (3).

We have assumed in the previous chapter that the set of alternatives and the set of agents are both finite. In this aspect, our discussions fall into the category of *discrete* social choice models, as opposed to *spatial* models which are extensively discussed in economics. In spatial models, the set of alternatives is thought of as an ordered set of points drawn from a continuum, and hence is infinite. This order is another contrast between discrete models and spatial models.

Although the set of alternatives is ordered in discrete models, the order is imposed for convenience of casting the votes or expressing the preferences in a systematic way. On the other hand, the order in spatial models is a natural order inherent from the underlying space.

3.2.3 Types of Preferences

The following is a categorization of preference-type

- Numerical preference structure is a structure for representing preferences using numeric values. The best example of numerical preference structures is a utility function $u : \mathcal{X} \rightarrow \mathcal{R}$ that takes as input a set of alternatives and produces as output a real number representing the utility of the input. This is why this type of preference structure is sometimes referred to as *utilitarian* preference. This category is also referred to as *cardinal* preference structure, especially when compared to ordinal preference structure in the same course of discussion.

- Ordinal preference structure is a structure for representing preference using order. The best example of ordinal preference structure is the preference relation discussed in Chapter 1.

- Qualitative preference structure is a hybrid structure for representing preferences using order on some mensurable scale (that could be, but is not necessarily, numerical). An example of this type is a function that takes as input a set of alternatives and outputs a set of a totally ordered (yet not numerical) scale.

While ordinal preference structures seem to naturally express preferences, numerical and qualitative structures are commensurable and allow a more precise expression of preferences. Moreover, numerical and qualitative structures allow comparisons between the preferences of different agents. Since the notion of the intensity of the preference is captured more in these structures, they allow inter- and intra agent preference comparison. This can be used to resolve some of the inconsistencies and paradoxes that arise when aggregating preferences.

Many voting systems implicitly devise a quasi-qualitative preference structure. For example, preferences in approval voting can be expressed as a preference relation where C_aPC_b, for all voters $I \in V$ and aIb and cId for all alternatives $a, b \in C_a$ and $c, d \in C_b$, where C_a is the set of alternatives approved by the agent and C_b is the set of alternatives not approved by the agent. Yet, the output is a total order on the set of alternatives where an alternative is placed based on the number of times it was preferred to some other alternative. Conversely, scoring protocols aggregate preferences that are based on points (natural numbers) given to the candidate, different combinations of input yield different outputs, however the output (with respect to the position of a given candidate) can be "ordered" from best to worst based on the position (number of points) of that candidate in the output.

A number of logic-based languages have been proposed for succinctly encoding preference relations over a set of alternatives. Propositional logic languages and weighted logics are two examples. Next, we exhibit some of these languages and discuss various criteria for considering a specific language for preference representation.

3.2.4 Preference Representation Languages

There are three main purposes of a preference representation language:

1. To encode the preferences efficiently, specially with respect to computational resources such as space.

2. To be used in the elicitation of preferences in case of partial preference description for example and

3. To be used in the aggregation of preferences into an outcome according to some defined rule.

Thereupon, the choice of a preference representation language is strongly linked to the preference elicitation process and to optimization of the preference aggregation output.

The criteria for choosing a PRL vary depending on the context and the underlying computational model. However, the following are some general basic criteria:

- **Compactness**. Compact preference representation languages that spare space are preferred especially in combinatorial domain where the set of alternatives or agents is very large and hard to trivially represent.

- **Complexity**. A preference representation language should be chosen such that the computational complexity of finding winner alternatives (and related problems such that checking whether an alternative is a winner and which of two alternatives is ranked higher) is considered. Lower complexity languages are, surely, more preferred.

- **Elicitation efficiency**. Elicitation efficiency (or elicitation-friendliness [25]) requires that a preference representation language is such that the design of algorithms used to elicit preferences and produce aggregate preference is easy.

- **Cognitive relevance**. A preference representation language should be close to the way agents naturally express their preferences in their language. It should be at least possible to simply derive the preferences of the agents from the language expressions without much complication.

- **Expressivity**. A preference representation language may have limitations in expression power. It may be not possible to represent some preference relations, utility functions or negotiation arguments using that language. Although this limitation can be acceptable depending on the (restricted) domain, generally a preference representation language with more expressivity is preferred.

- **Comparative succinctness**. Succinctness and compactness (of a preference representation language) are usually used as synonyms in the literature. However, comparative succinctness implies the relative succinctness of one language in comparison to another. Given two preference representation languages L_1 and L_2, the question is: can every preference expressed in L_1 also be expressed in L_2 using "comparable" (i.e., within polynomial factor) resources?

- **Comparative expressivity**. Similarly and more simply the question here is: can every preference expressed in L_1 also be expressed in L_2? Preference representation languages that are easily translated to other languages are considered a better choice because of their flexibility and applicability to difference domains (via translation).

The reader may have noticed the difficulty in satisfying all these criteria by one language. For example, compactness might come at the price of less cognitive relevance.

Examples of Preference Representation Languages

Preferences can be expressed in several ways. A specific representation language may focus on some set of aspects to capture the notion of preference in a given domain. The following is a brief presentation of some PRLs:

- Goal importance and compensation languages, including the following two examples:

  - $R_{penalties}$: a frequently used preference representation language that focuses on goals, weights of goals and penalties associated with goals if not satisfied. An agent expresses its preferences in terms of propositional formulas to be satisfied. Each formula is associated with a number (the weight) that indicates the importance of satisfying that formula. Alternatives are compared based on their weights.
  - R_H: as a refinement to the previous rule which differentiates between alternatives merely based on whether they satisfy goals or not, closeness to goals can be introduced as a criterion for accepting alternatives. In other words, an agent prefers an alternative close to the goal over an alternative that is far from the goal. This closeness is usually measured using the Hamming distance. The exact interpretation of the Hamming distance depends on the domain[4].

- Prioritized Goals languages: The first two languages are based on the idea of goal importance and compensations among goals. This means that violating one very important goal can be compensated for by satisfying a number of less important goals. When such compensations are not possible, languages based on the idea of prioritized goals are suggested (e.g., $R^{bestout}_{prio}$, $R^{discrimin}_{prio}$, and $R^{leximin}_{prio}$). This is the unified feature of prioritized goals languages. The basic difference between these languages is how goal-priority is introduced as an extension to a preference relation. The detail on these extensions in their rather involved knowledge representation scope is explained elsewhere [14].

- Conditional logics: in this type of languages, preference relations are constrained by some conditions (according to the definition of the language), only preference orders satisfying these conditions are considered when comparing a set of alternatives (e.g., R^{cond}_{S} the standard preference relation and R^{Z}_{cond}.).

- Ceteris Paribus (CP) preferences language: In a Ceteris Paribus ("all else being equal") language (e.g., proposed in [32]), preferences are defined in a multicriteria decision making (MCDM) setting. In CP languages, preferences are expressed with respect to one criterion (attribute or dimension) while holding all others constant. Typically, a preference in this language is a propositional logic statment like: "all other things being equal, I prefer these alternatives over these other ones." The "other things" is translated, in the expressions of this language, as variables not included in the variable-set of the propositional formulas expressing preferences.

We now use the previous examples of preference representation language to touch on a concrete realization of some of the abstract criteria for selecting a PRL, namely comparative succinctness and comparative expressivity. The following table summarized the results obtained in [14] regarding translation among these languages. The table is to be read in light of the following:

No means the translation is impossible using polynomial space. The type of proof used for obtaining this result is:

[4] For an abstract definition see [14].

1. a direct argument based on cases where the second language is limited in expressivity when completed to the first (i.e., there are some ordering that the first language can represent while the second cannot.). Since the second language cannot represent some preferences in the first place, these preferences cannot be translated to preferences in this language using efficient space resources.
2. a direct argument based on cases where the first language can represent some orderings in polynomial space where the second language needs at least suprapolynomial space to express those orderings.

c-No stands for "conditional no" and means that a polynomial *time* translation is not possible unless the polynomial hierarchy collapses.

w-No stands for "weak no" and means there is no polynomial *space* translation unless we relax the condition of requiring the formulae of the resulting language to be in Conjunctive Normal Form (CNF) since requiring this and the possibility of a polynomial size translation combined together would violate results from circuit complexity[5].

Table 3.1: Summary of results on translation among preference representation languages.

from＼to	R_{pen}	R_H	$R_{prio}^{bestout}$	$R_{prio}^{leximin}$	$R_{prio}^{discrimin}$	R_{cond}^S	R_{cond}^Z	R_{CP}
R_{pen}	Yes	Yes	No_2	?	w-No	No_2	No_2	No_2
R_H	c-No	Yes	No_2	c-No	w-No c-No	No_2	No_2	?
$R_{prio}^{bestout}$	Yes	Yes	Yes	Yes	Yes	No_2	Yes	Yes
$R_{prio}^{leximin}$	Yes	Yes	No_2	Yes	w-No	No_2	No_2	No_2
$R_{prio}^{discrimin}$	No_1	No_1	No_1	No_1	Yes	No_2	No_1	No_2
R_{cond}^S	No_1	No_1	No_1	No_1	?	Yes	No_1	?
R_{cond}^Z	Yes	Yes	Yes	Yes	Yes	No_2	Yes	Yes
R_{CP}	No_1	No_1	No_1	No_1	w-No	No_2	No_1	Yes

3.2.5 Single-peaked Preferences

Consider three individuals who plan to buy houses. Assume that the first person has a small budget for buying a house, the second person has a medium budget and the third person has a big budget. Also, suppose that there are three types of houses, small-sized low-priced house, medium-sized medium-priced house, and a spacious expensive house. We expect the person with the small budget to prefer a small-sized house to a medium-sized house and to prefer a medium-sized house to a large one. The person with the big budget will have an inverse preference ordering. The person with the medium budget will probably prefer a medium size house to both small and big houses. We further assume that all other factors

[5]For example, the existence of such a translation contradicts the impossibility of expressing the majority function with a CNF circuit of size that is polynomial in the number of variables.

are held constant or that the price of the house according to its size is the only criterion considered.

The preferences of these three individuals are *single-peaked*. Each person has a "peak" or an "ideal point" of preference and on both sides of the peak—unless the peak is an extreme point of the alternative set—alternatives are less desired. The further a point (representing an alternative) from the peak; the less preferred that alternative is. Figure 3.5 illustrates this situation.

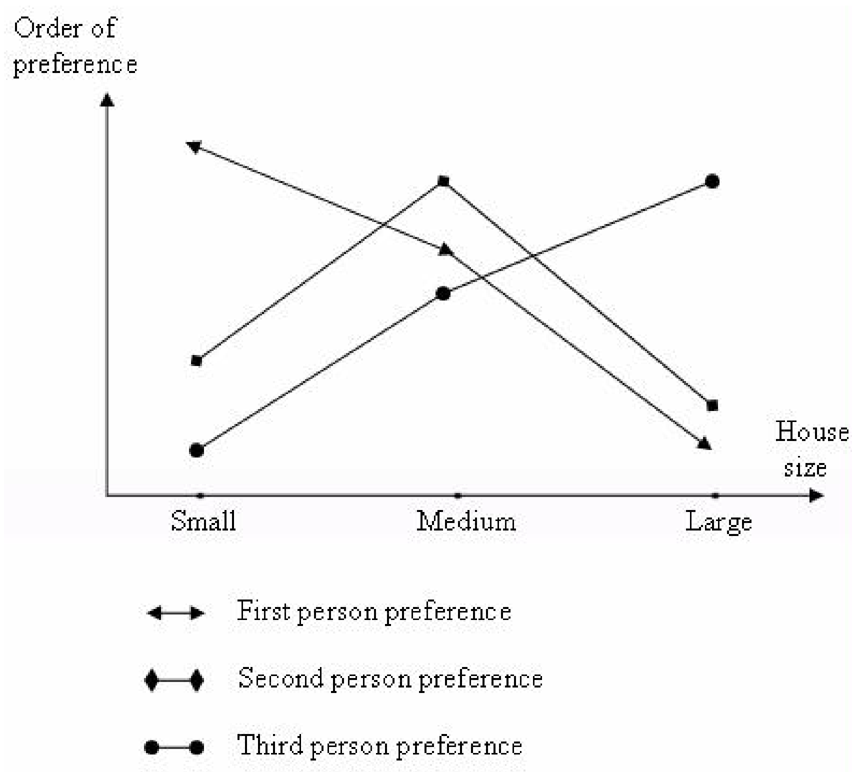

Figure 3.5: Three single-peaked preferences.

It is important to note that the ordering of the alternatives on the horizontal axis is key here. If we change that ordering, the picture might change totally and preferences, although the same, become no longer single peaked. Also, it is worth mentioning that this order is independent of any preference and is merely introduced to structure the preferences.

The absolute height of the curves at any point is also irrelevant. The important feature is the relative height and overall shape of the curves in comparison to one another. Nevertheless, the absolute "utility" of the alternative can be introduced and represented by the vertical axis in numerical representation structures. Also, the continues lines between the points are drawn only to assist the human eye in recognizing the shape of the curve and therefore the preference behavior. These lines do not necessarily indicate a continuous set of alternatives. Only the relative position of a preference at each point of the set of alternatives is significant. Hence, the term "Point Set" on the diagrams.

Single peakedness is a property of preferences that makes them escape some voting paradoxes. Since ingle-peakedness also expresses the intensity of preference, it is possible to have a cycle in the majority graph, yet with a possible ordering of the alternatives involved in the cycle. This is because different intensities of preferences may contribute to unequal weights on the edges, hence placing an order on the alternatives. Although there are other restrictions on preferences that circumvent the voting paradox and impossibility results, single-peakedness seems to be a well known and natural restriction.

In single-peaked preference settings, the alternatives are ordered on a line from left to right in sequence (or right to left[6]) and the positions of alternatives on this line represent their positions on a specific matter. For example, in presidential elections candidates can be positioned in an ideological spectrum that ranges from most liberal to most conservative where the position of a candidate on that spectrum is represented by their positions on the liberal-conservative spectrum.

An **ideal point** for a voter is the most preferred position, alternatives closer to the ideal point are preferred to alternatives that are further away, in other words, the further an alternative from a voter's ideal point the less support this alternative gets from that voter.

For example, if the set of alternatives is $\{u, v, w, x, y, z\}$, and they are positioned on a line from left to right in the following order: x, v, y, z, u, w then the vote $z > y > v > u > w > x$ is single-peaked (around z), but the vote $z > x > v > u > w > y$ is not (with respect to z as an ideal point) since x is further away from z than v yet x is preferred to v.

Definition 3.2.1 (Condition of Single-peakedness)**.** *A profile $\{R_1, R_2, \ldots, R_n\}$ of individual preferences satisfies the property of single-peakedness if there exists a strong ordering S such that for all $i \in V$, xR_iy and $B(x, y, z)$ imply yP_iz, where $B(x, y, z)$ means that y is between x and z in the defined sequence of alternatives, and x, y and z are distinct elements of the set of alternatives. $B(x, y, z)$ can happen if either x is followed by y and y is followed by z in the sequence of alternatives or z is followed by y and y by x in the sequence [17].*

The main appeal to single-peaked preferences is the following *possibility* theorem:

Theorem 3.2.1 (A Possibility Theorem for Single-peaked Preferences)**.** *Provided the number of voters[7] is odd, the majority decision rule is a social welfare function[8] for any number*

[6] The alternatives are ordered from left to right or from right to left. The direction does not matter as long as it is well defined with respect to a reference direction and as long as all voters' preferences are drawn with the respect to a fixed ordering of the alternatives on the reference direction.

[7] In some publications (e.g., [17]), the text of the theorem states "concerned voters." The set of concerned voters is the set of voters who are not indifferent between every pair of alternatives in a given set (or subset) of alternatives to decide upon. When all preferences are strict and total, all voters are concerned voters; no voter abstain from voting and no vote is incomplete or indifferent among two alternatives.

[8] Here, the majority decision rule is a social welfare function since it meets the requirements of fairness and uniformity. It satisfies the conditions of a social welfare function that Arrow stipulated in his impossibility result. Refer to Chapter 1 of this thesis for details.

of alternatives if the voters' individual preferences are all single-peaked over each triple of alternatives [17].

This theorem states that single-peakedness of preferences (with the oddness requirement of the number of voters) is a sufficient condition for the outcome of the majority social welfare function to be an *ordering* (i.e., transitive and complete). However, this theorem says nothing about single-peakedness being a necessary condition for the outcome of a social welfare function to be an ordering. Indeed, single-peakedness is not a necessary condition in this context. It is possible to find cases where preferences are not single peaked yet the result of a social welfare function is an ordering over the alternatives.

A mirror concept to single-peakedness is single-cavedness (or single-troughedness). In single-peakedness, a voter has a most preferred point and the further an alternative from this ideal point the less preferred that alternative is to the voter. In single-cavedness, the converse is true, a voter has a least preferred point and the further an alternative from this point the more preferred that alternative is to the voter. In the budget-house example, if the order on the horizontal line is reversed, the key representing voters is reversed and the second person's curve is flipped upside down we get a figure depicting single-caved preferences. Generally, and unless the ideal point is at one end of the preference sequence, a single-peaked preference curve has a $\cup$-shape and a single-caved preference curve has a $\cap$-shape. The results applying to single-peaked preferences also apply to single-caved preferences.

Single-peakedness of preferences applies to *unidimensional* models (discrete or continuous). A similar concept (yet not exactly the same) called **convexity** is defined in the context of multidimensional models where preferences are not represented by curves but by surfaces and each dimension represents a distinct issue.

Recognizing Single-peaked Preferences

Testing for single-peakedness of preferences is a nontrivial task. First, it is not easy to determine preferences in the first place. This applies to ombinatorial domains as well as to popular elections for practical reasons, such as the difficultly and cost of drawing a reliable sample from a large population[9]. Second, even if preferences are known then recognizing single-peakedness amounts to finding a permutation of the candidates with respect to which the preferences of all voters are single-peaked. This is an issue, since for m candidates the total number of permutation is $m!$. As the number of candidates increases, the total number of permutations becomes prohibitively large. In many natural settings, the number of candidates itself can be relatively large. For example, in 1986 *The New York Times* of April 1st reported twenty candidates for mayor of Tulsa, Oklahoma! And 20! is equal to 2432902008176640000.

However, when placing some conditions on the voting setting and under other properties of the preferences (see the Stable Matching example below), single-peakedness can be

[9]Note that this property needs to be tested before the election takes place to exploit this structure in the process of determining winners. A census for determining preferences *a priori* is therefore out of question since this census may as well record the votes also.

tested for in polynomial time, sometimes as efficient as quadratic time in the number of candidates [2].

If all possible permutations of alternatives are exhaustively considered and for each permutation, the preferences are not single-peaked with respect to that sequence of the alternatives, then those preferences do not enjoy the property of single-peakedness. Nevertheless, if a single permutation of the alternatives does not yield single-peaked preferences, then we cannot draw conclusions about the global picture of the preferences. Figure 3.6 shows a matrix example of preferences that are not single-peaked and further more exhibiting a cyclic majority. Note that reading down any column (or across any row), we cannot find an non-dominated alternative.

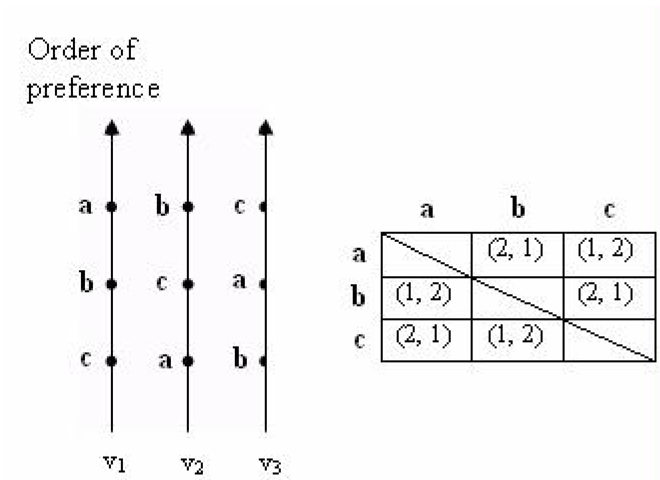

Figure 3.6: Condorcet Paradox with non-single-peaked preferences.

Implications and Results

The following list contains some important results applying to single-peaked preferences:

- When agents/voters report their preferences truthfully and an alternative a wins under the majority rule, no other alternative can be made the winner by misrepresenting the preferences of some set of voters. However, it is possible to strategically vote such that no alternative wins under majority rule [3]. This implies that when preferences are single-peaked, voters have no incentive to misrepresent their preferences in order to make a non-winning alternative, the winner under majority rule.

- The *Median Vote Theorem*, attributed to Black [4] states single-peakedness as a condition for majority rule to produce transitive social ordering and specifies the value of the pairwise majority election winner:

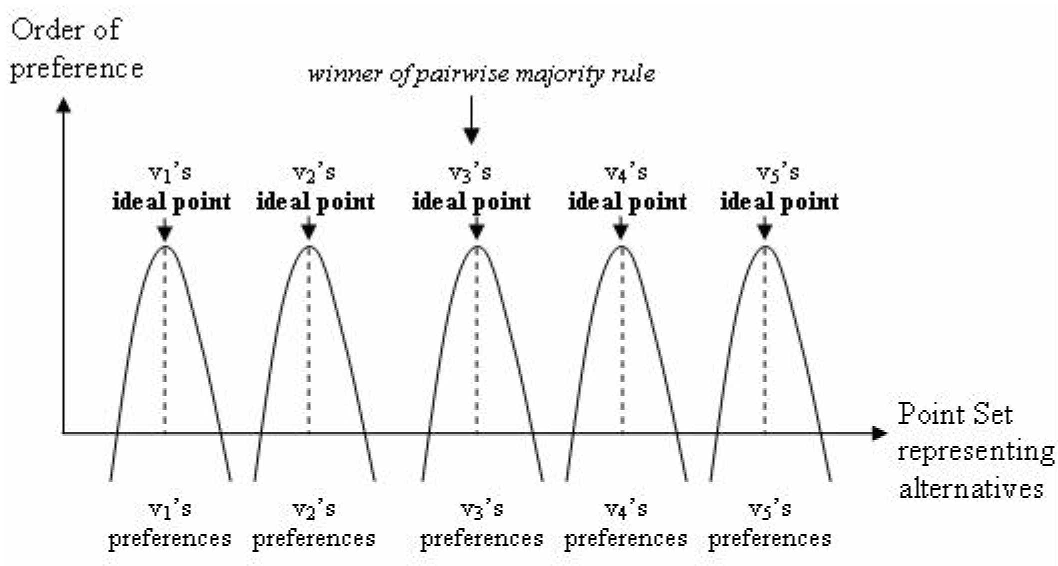

Figure 3.7: Median Vote Theorem (for 5 voters).

Theorem 3.2.2 (Median Vote Theorem)**.** *[20] If the number of voters is odd, and the preferences are single-peaked (in one dimension) then:*
1. The median of the voters' ideal points is a Condorcet winner, and
2. majority rule leads to a transitive social ordering.

- If either the order with respect to which preferences are single-peaked is known or the complete preference of at least one other agent is known then single-peaked preferences can be elicited[10] using only a linear number of comparison queries. Also, a sublinear number of queries does not suffice to elicit single-peaked preferences [12].

Besides efficient preference elicitation, single-peaked preferences have been studied in other domains and applications. For example, it has been shown that if preferences are single-peaked and "narcissistic" (i.e., an alternative is its own ideal) then there exists a unique stable matching [2]. The Stable Matching (or Stable Roommates) problem is to match $2n$ persons in pairs so that no two prefer each other to their assigned partners. Furthermore, when preferences obey single-peakedness and narcissism then the stable matching can be constructed in $O(n)$ steps using a simple iterative algorithm [2].

Recently, the following characterization of single-peaked preferences was obtained [1]: A preference profile is linearly ordered over the set of alternatives such that these preferences are single-peaked with respect to the linear order L if and only if it satisfies these two properties:
1. For any subset of alternatives the set of alternatives considered as the worst by all agents cannot contain more than two elements.

[10] The next section is dedicated entirely to the topic of preference elicitation.

2. Two agents cannot disagree on the relative ranking of two alternatives with respect to a third alternative but agree on the (relative) ranking of a fourth one.

3.3 Preference Elicitation

3.3.1 Incomparability and Incompleteness of Preferences

In some voting scenarios, a decision based on *partial* information about preferences needs to be made. Partial information about preferences can refer to an incomplete collection of complete preference profiles, or to a complete collection of *partial* preferences. The first can occur when the preference-elicitation process is not complete at the time when outcomes must be determined. The second for example, can occur when a voter does not specify her preferences between two alternatives or when a voter states that these alternatives are *incomparable*. In all such cases, voting rules (or preference aggregation methods in general) face the challenge of having to determine outcomes given incomplete preferences.

When the collection profile of a group of agents does not include the preference profiles for some agents or when some agents do not state their preferences among a subset of alternatives then we are before a case of *incomplete* or *partial* preferences. Recall that two alternatives a and b can be related in a preference profile by one of the following: aRb, aPb, or aIb standing for a weak preference, a strict preference, or an indifference relation respectively. Succinctly, let us use the symbols of the set $\{\geq, >, =\}$ to stand for the above-mentioned relations in order.

In Multiple Criteria Decision Making (MCDM), when an agent is presented with two alternatives, the agent might refuse to compare the alternatives since the criterion considered for each alternative is different. For example, when selecting the next step of a resource allocation plan, an agent might find an easy to implement less efficient step incomparable to a more involved efficient one since there are two distinct criteria (implementation difficulty and efficiency) to be considered (to simplify the idea of incomparability, think of the following analogy: a human—choosing between two computers—might find the comparison between a slow cheap computer and a fast expensive one less obvious since there are two distinct criteria (speed and cost) to be considered. He might readily compare machines based on their prices only or speed only.). When considering incomparability as a possible way of describing preferences among alternatives, we can extend our set of preference symbols to $\{\geq, >, =, \sim, ?\}$ where $\sim$ stands for "incomparable" and ? for "unspecified."

There are a number of situations in which a preference aggregation method needs to aggregate incomplete preferences. For example:

- When some k voters have not reported their preferences yet, while others have. Then the collection of preference profiles R^n contains k complete preference profiles of k voters and $n - k$ empty preference profiles.

- When all voters have voted but an individual preference profile is not necessarily a complete preference relation. This is possible in a number of cases:

  - If the votes are cast incrementally, for example, in a multistage (or multi-round) voting rule or preference aggregation method in general. This is also similar to implicitly eliciting votes in a multistage rule after eliminating a subset of candidates. In these cases, at each stage of the vote aggregation processes not all information about preferences or about candidates' status is available,
  - when a voter does not have enough information yet to compare two alternatives, so the choice between these alternatives is left unspecified in the preference profile of that voter, or
  - when a voter does not want to express its preference over a subset of alternatives either for privacy reasons or when the alternatives are incomparable according to the criteria the voter uses to evaluate these alternatives.

- When all voters have cast their votes but a new set of candidates is introduced and merged with the original set. The preferences of voters regarding this new subset are unknown and the result is incomplete preferences with respect to the new (union) set of candidates.

- When the preference elicitation process terminates and decisions need to be made based on the—possibly partially—elicited preferences.

- When preferences are expressed in a compact representation language (such as CP-nets [7]) that generates partial preference relations in general.

The preference relation representing partial preferences is therefore not complete, as a result of unspecified preferences and incomparable alternatives, and it is possibly reflexive and symmetric, as a result of indifference and incomparability between alternatives. Such a preference relation is referred to as a *partial* ordering.

3.3.2 What is Preference Elicitation?

The topic of preference elicitation has gained wide attention especially in the domain of combinatorial auctions and bid elicitation (e.g., [19, 11]). It has also been studied for exchange and negotiation as well (e.g., [9]). Compared to this wide attention to preference elicitation in some preference aggregation methods, less attention has been given to *vote* elicitation, where the traditional practice and typical applications require complete preferences as input. However since voting has been recently suggested in various domains for tackling different problems, as we will see in a later chapter, extending voting rules to handle incomplete preferences and specifically discussing *vote elicitation* as a special case of preference elicitation have emerged as new topics in the voting literature. In this section we explore some aspects of *vote* elicitation. Since voting is one way of reporting preferences, we first discuss preference elicitation in general. Preference elicitation consists of interacting with agents in order to gather information about their preferences. This information is used to answer the following question:

Can outcome of the preference aggregation method be determined without the need for any further information?

This entails that preference elicitation must also answer the implicit question:

What question(s) must be asked to which agents in order to *elicit* the right information (i.e., the information needed for determining the outcome)?

These questions are usually referred to as "queries" in the preference-elicitation literature. Related questions are: What type of queries is to be asked? How many queries can the elicitor ask an agent in each step of the elicitation process and in general? What is the computational complexity of this process? And when to terminate it safely (i.e., when is the outcome surely known, the case in which further elicitation is useless since any additional information will not change the current outcome)?

There are different types of preference elicitation:

- Full elicitation: the entire preference profile is obtained from every agent (once).
- Coarse elicitation: every time the elicitor interacts with an agent, the elicitor asks for the agent's preferences.
- Fine elicitation: in each query to an agent, the elicitor asks a specific question about the agent's preferences, for example, which of two alternatives the agent prefers more.

The previous notions of preference elicitation also apply to vote elicitation. The definitions can be easily rewritten to define vote elicitation by simply replacing "preference aggregation method" with "voting protocol," replacing "preference" with "vote" and replacing "agent" with "voter."

3.3.3 Vote Elicitation

In traditional voting settings, a voter is usually asked for his complete preferences, whether these preferences are expressed as a total ordering, as approvals or as a unique favorite choice. Although some voters may abstain from voting over a specific set of alternatives or in some domains such as presidential elections, some voters would not register to vote, as long as a voter is to engage in the voting process, he must report his preferences (over the set of alternatives) as specified by the voting rule in use. It is possible, however, that not all votes on the election level or not all preferences of a single voter at the voter level are needed for determining the final outcome. For example, if more than half of the votes are considered and they do not place candidate a over candidate b, then we know that a is not the winner of the majority voting rule without considering the remaining votes. Also, if only the top ranked candidate is relevant for a specific voting rule then the entire preference of each voter over all candidates is not needed. Yet another example of cases where complete information about preferences is not needed is when *enough* information has been gathered to conclude that a given alternative does not have a chance to win. Nonetheless, in some scenarios (such as the ones mentioned early in this section), incomplete preferences are all that is input to a voting rule that needs to elicit more preferences before being able to compute the outcome.

Yet, most theoretical treatment of voting procedures, especially axiomatic foundation of some impossibility results and complexity-theoretic treatments seeking hardness results, assume complete preferences and in some problems complete knowledge available for each voter about all other voters' preferences. For if hardness results are established under this condition, hardness readily follows for more obscure settings in which not all preferences are known.

Nevertheless, forms of vote elicitation do take place in real life voting, in order to predict outcomes based on partial information and/or to tactically influence the voting process and therefrom the outcome.

Vote elicitation can empower the voting protocol to determine the outcome of the election before the process of casting all the votes is complete. This can also reduce both the computational complexity and the practical effort involved in determining preferences and outcomes. In addition to this, some voters prefer not to reveal all of their preferences for privacy reasons and also to discourage other agents from voting strategically based on information about other voters' preferences.

Therefore, voting rules under incomplete preferences, which is closely related to relevant and necessary information for determining outcome—which is the information sought via vote elicitation—have presented themselves as imperative topics of study in the voting literature.

3.3.4 Extending Voting Rules to Handle Incomplete Preferences

Consider the Condorcet rule with the set of three alternatives $\{a, b, c\}$, three voters $\{i, j, k\}$, and a social preference profile $R^n = (a > b > c; a?b > c, a > c; a \sim b > c, a < c)$.

The preferences of each voter (except the last one in the list) are followed by a semicolon to separate them from the preferences of the next voter. As mentioned earlier, $\sim$ stands for "incomparable" and ? for "unspecified." The preference profile R^n above denotes the preferences of three voters which are as follows:

Voter i prefers alternative a to b, alternative b to c, and alternative a to c (by transitivity).
Voter j prefers alternative b to c and alternative a to c, but does not specify his preference between alternatives a and b.
Voter k finds alternatives a and b incomparable, he prefers alternative b to c and alternative c to a.

The following are possible extensions (completions) of this profile:
$R^n_{e1} = (a > b > c; a > b > c; a \sim b > c, a < c)$,
$R^n_{e2} = (a > b > c; a \geq b > c; a \sim b > c, a < c)$,
$R^n_{e3} = (a > b > c; a < b > c; a \sim b > c, a < c)$,
$R^n_{e4} = (a > b > c; a \leq b > c; a \sim b > c, a < c)$, and
$R^n_{e5} = (a > b > c; a = b > c; a \sim b > c, a < c)$.

In general, there are exponentially many extensions of a partial preference profile. Different approaches have been suggested to extend voting rules in order to apply them to incomplete preferences:

- Consider all extensions of R^n and apply $\mathcal{E}$ to each and every extension, gather the results and if necessary aggregate the results using the same or some other method. In the previous example, the set of possible outcomes is $\{a, b\}$.

- Consider a subset of extensions (typically a singleton) using some completion process, and apply $\mathcal{E}$ to the resultant complete profile.

- Generalize the definition of $\mathcal{E}$ so that it can input incomplete preferences. The new definition should output the same result as the original when the preferences in the input are complete.

Some related work [21] focused on the first approach perhaps because it is natural when incompleteness is intrinsic in the preferences and there is no other way to decide which extension to consider. However, considering a single extension has also been suggested using different methods such as candidate gravitation towards preferences [6] or exploiting indifferences [31].

3.3.5 Effective Elicitation

Preference elicitation ought to be effective. Effective preference elicitation is such that the right outcome is found by eliciting minimal information about preferences. A preference elicitation process that needs to obtain information about all or almost all preferences is useless since it may as well be omitted and replaced with a single deterministic approach for fully obtaining preferences and drawing outcomes (which would be the actual election).

Effective preference elicitation makes use of information about the candidates, information about the voters and information about how agents relate to each other and to candidates. Eliciting preferences effectively implies knowing when to safely terminate the elicitation process, which in turn implies knowing what preferences to elicit from which voters. Both these elements of effective elicitation are formally defined in [13] as follows:

Definition 3.3.1 (Elicitation-Not-Done)**. Given***: A voting rule $\mathcal{E}$, a set of votes V, a number k of votes that are still unknown and a candidate c.*
Question*: Is there a way to cast the k votes so that c will not win the $\mathcal{E}$-election?*

Definition 3.3.2 (Effective-Elicitation)**. Given***: A voting rule $\mathcal{E}$, a set of votes V and a number k.*
Question*: Is there a subset of V of size $\leq k$ that decides the outcome of the $\mathcal{E}$-election* (i.e., the winner(s) will be determined based on the preferences of these voters only regardless of the preferences of other voters)*?*

It turns out that for many known voting rules, both Elicitation-Not-Done and Effective-Elicitation are NP-complete. We will revisit these results in the next chapter.

3.3.6 Possible and Necessary Winners

A possible winner is a winner in (at least) one of the completions of an incomplete preference profile. A necessary winner is a winner in all completions of an incomplete preference profile. When the preference profile is complete, the set of possible winners and the set of necessary winners coincide. Formally,

Definition 3.3.3 (Possible and Necessary Winners)**.** *Let $\mathcal{E}$ be a voting rule, C a set of alternatives, R^n a multiset of (possibly incomplete) preference profiles, and let $Ext(R^n)$ stand for the set of possible extensions of R^n into a multiset of complete preference profiles.*

- *$x \in C$ is a necessary winner for R^n with respect to $\mathcal{E}$ if and only if for all $\mathcal{T} \in Ext(R^n)$, $x \in \mathcal{E}(\mathcal{T})$. The set of necessary winners under $\mathcal{E}$ given R^n is denoted by $NW_{\mathcal{E}}(R^n)$.*
- *$x \in X$ is a possible winner for R^n with respect to $\mathcal{E}$ if and only if there exists a $\mathcal{T} \in Ext(R^n)$ such that $x \in \mathcal{E}(\mathcal{T})$. The set of possible winners under $\mathcal{E}$ given R^n is denoted by $PW_{\mathcal{E}}(R^n)$.*

For any voting rule $\mathcal{E}$, the following properties hold:

- For all R^n, $NW_{\mathcal{E}}(R^n) \subseteq PW_{\mathcal{E}}(R^n)$.
- For all R_1^n, R_2^n such that $R_1^n \subseteq R_2^n$, $NW_{\mathcal{E}}(R_2^n) \subseteq NW_{\mathcal{E}}(R_1^n)$ and $PW_{\mathcal{E}}(R_2^n) \subseteq PW_{\mathcal{E}}(R_1^n)$.

Since there can be exponentially many extensions of a partial preference profile, there is no general polynomial time algorithm for computing necessary and possible winners for every voting rule (unless P = NP). However, for some families of voting rules it is possible to determine necessary and possible winners in polynomial time. When the elicitation process starts, the set of necessary winners is empty and the set of possible winners contains all alternatives (unless further information is initially known such that to restrict the set of possible winners to a proper subset of alternatives). As more preferences are elicited, the set of necessary winners expands and the set of the possible winners shrinks. The elicitation process is done when the two sets coincide.

Proposition 3.3.1 (Termination of Vote Elicitation)**.** *Given a voting rule $\mathcal{E}$ and a partial preference profile R^n, the vote elicitation process is over if and only if $PW_{\mathcal{E}}(R^n) = NW_{\mathcal{E}}(R^n)$.*

This is because when the two sets coincide, there is enough information to declare the winners. Additional information will not possibly add new winners since winners were obtained from the set of all possible winners which was initialized to the set of *all alternatives*. Therefore, the remaining preferences can be safely disregarded.

If the voting rule satisfies Independence of Irrelevant Alternatives (IIA)[11], then determining whether an outcome $a \in PW - NW$ is a winner or not, can be archived by asking the voters for their preferences between a and all other alternatives b in pairwise comparisons. Therefore, preference/vote elicitation for rules satisfying IIA can be done efficiently in polynomial time.

[11] The condition of IIA is defined in Chapter 1.

3.3.7 Vote Elicitation and Manipulation

If an agent wants a specific alternative to win and plans to (strategically) vote in favor of this alternative then the agent must at least determine whether this alternative is a *possible* winner. Hence, the close relation between the problem of determining possible and necessary winners and the manipulation problem. This relation also stems from the fact the elicitation itself is closely related to the computation of possible and necessary winners, and this elicitation process can create opportunities for both the elicitor and agents to strategically influence the outcome based on knowledge about (others') preferences. For example, in multiagent settings, an agent can infer the number of agents elicited so far by comparing the time at which the elicitation process started and the time at which that agent was queried. The time difference together with the query the agent is asked can be used to determine (or approximate) the number of agents elicited so far and the nature of their responses.

General manipulability results concerning social choice functions (which include voting rules by definition[12]) on partial preferences have been obtained in [30], [29], and [27]. We need to define the following terms before presenting those results[13]:

Definition 3.3.4 (Strong dictator)**.** *A strong dictator is a voter such that, no matter how the others vote, this voter's ordering/choice is the outcome.*

Definition 3.3.5 (Dictator)**.** *A dictator is a voter such that, no matter how the others vote, no outcome is selected outside the choice set of this voter.*

Definition 3.3.6 (Weak dictator)**.** *A weak dictator is a voter such that, no matter how the others vote, some choices of this voter will always be included in the outcome.*

Given these definitions, the following are results concerning manipulability of social choice functions (and hence voting rules), under partial preferences:

Theorem 3.3.1. *If we have at least two agents and at least three alternatives, a social choice function on partial order without ties that is unanimous and monotonic has at least one weak dictator [29].*

Theorem 3.3.2. *If a social choice function is strategy-proof and onto then it is unanimous and monotonic [27].*

Theorem 3.3.3. *If a social choice function is strategy-proof and onto then it has at least one weak dictator*[14] *[27].*

Proof. From theorems 3.3.1 and 3.3.2 above. □

[12] Consult Chapter 1 of this thesis for these definitions.

[13] There are alternative definitions of these terms based on pairwise comparisons between the alternatives [27].

[14] This result extends the Gibbard-Saterthwaite theorem, which is discussed in some detail in the next chapter, to hold for social *choice* functions on *partial* preferences.

3.3.8 Some Results

We conclude this section with a list of results on vote elicitation and related problems:

Proposition 3.3.2. *Given a voting procedure $\mathcal{E}$ that is computable in polynomial time and a partial (i.e., an incomplete) preference profile, the vote elicitation process is over if and only if $PW_{\mathcal{E}} = NW_{\mathcal{E}}$.*

Theorem 3.3.4. *For Approval, STV, Borda count, Copeland and Maximin voting rules,* Effective-Elicitation *is* NP*-complete [13].*

Theorem 3.3.5. *In general, determining whether $x \in PW_{\mathcal{E}}(R^n)$ is* NP*-hard.*

Theorem 3.3.6. *In general, determining whether $x \in NW_{\mathcal{E}}(R^n)$ is* coNP*-hard.*

Theorem 3.3.7. *For a polynomially-computable voting rule $\mathcal{E}$, determining whether $x \in PW_{\mathcal{E}}(R^n)$ is in* NP.

Theorem 3.3.8. *For a polynomially-computable voting rule $\mathcal{E}$, determining whether $x \in NW_{\mathcal{E}}(R^n)$ is in* coNP.

Theorem 3.3.9. *For positional scoring protocols, possible and necessary winners can be computed in polynomial time [21].*

Theorem 3.3.10. *For STV, computing possible winners is* NP*-complete and computing necessary winners is* coNP*-complete [28].*

Theorem 3.3.11. *It is* NP*-hard to return a superset of the possible winners PW^* under STV such that we guarantee $\|PW^*\| < k\|PW\|$ for some given positive integer k [28].*

Theorem 3.3.12. *Possible and necessary Condorcet winners can be computed in polynomial time [21].*

Theorem 3.3.13. *For a given voting procedure, if necessary and possible winners are computed in polynomial time then deciding whether there is a (constructive/destructive) manipulation[15] is also possible in polynomial time [21].*

Theorem 3.3.14. *If a voting rule $\mathcal{E}$ satisfies IIA and is computable in polynomial time, then determining the set of winners via preference elicitation is polynomial in the number of agents and alternatives [28].*

3.4 Comments and Bibliographic Notes

- This chapter discussed representation and elicitation of preferences. In preference representation, different approaches were explained briefly. For more on the capabilities and limitations of ordinal approaches to group decision-making see [18, 15]. A noteworthy amount of work have elaborated on logical representation of preferences, for example [22, 16, 23]. Further more, development of logical calculus for representing and reasoning with preferences has been presented in [6].

[15]The complexity of manipulation will be discussed in some detail in the next chapter.

- Preference elicitation methods were generally surveyed in [10]. One of the approaches to preference elicitation is the constraint-based approach which has not been discussed in this chapter, the reader is referred to [8] for more on this.

- Another topic related to preference elicitation, and generally to AI aspects of computational social choice and decision making, is the topic of query learning (e.g., [5, 33, 24]). Work in this direction combines preference elicitation and computational learning theory (in specific, learning via queries). This work, however, is more pertinent to preference elicitation in combinatorial auctions.

Chapter 3 Bibliography

[1] Ballester, M., and Haeringer, G. A characterization of single-peaked preferences. UFAE and IAE Working Papers 656.06, Unitat de Fonaments de l'Anàlisi Econòmica (UAB) and Institut d'Anàlisi Econòmica (CSIC), 2006. Available at: `http://ideas.repec.org/p/aub/autbar/656.06.html`.

[2] Bartholdi,III, J., and Trick, M. Stable matching with preferences derived from a psychological model. *Operations Research Letters 5*, 1 (1986), 165–169.

[3] Black, D. On the rationale of group decision-making. *The Journal of Political Economy 56*, 1 (1948), 23–34.

[4] Black, D. *The Theory of Committees and Elections*. Cambridge University Press, Cambridge, 1958.

[5] Blum, A., Jackson, J., Sandholm, T., and Zinkevich, M. Preference elicitation and query learning. *Journal of Machine Learning Research 5* (2004), 649–667.

[6] Boutilier, C. Toward a logic for qualitative decision theory. In *KR' 94: Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning* (1994), J. Doyle, E. Sandewall, and P. Torasso, Eds., pp. 75–86.

[7] Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., and Poole, D. Preference-based constrained optimization with CP-nets. *Computational Intelligence 20*, 2 (2004), 137–157.

[8] Boutilier, C., Brafman, R., Geib, C., and Poole, D. A constraint-based approach to preference elicitation and decision making. In *Working Papers of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning* (Menlo Park, CA, USA, 1997), J. Doyle and R. H. Thomason, Eds., American Association for Artificial Intelligence, pp. 19–28.

[9] Buffett, S., and Spencer, B. Learning opponents' preferences in multi-object automated negotiation. In *EC' 05: Proceedings of the 7th International Conference on Electronic Commerce* (New York, NY, USA, 2005), ACM Press, pp. 300–305.

[10] Chen, L., and Pu, P. Survey of preference elicitation methods. Technical report, Swiss Federal Institute of Technology in Lausanne, Lausanne, Switzerland, 2004.

[11] Conen, W., and Sandholm, T. Preference elicitation in combinatorial auctions. In *EC' 01: Proceedings of the 3rd ACM Conference on Electronic Commerce* (New York, NY, USA, 2001), ACM Press, pp. 256–259.

[12] Conitzer, V. Eliciting single-peaked preferences using comparison queries. In *AAMAS' 07: Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems* (Honolulu, HI, USA, 2007), pp. 408–415.

[13] Conitzer, V., and Sandholm, T. Vote elicitation: Complexity and strategy-proofness. In *AAAI' 02: Proceedings of the 18th National Conference on Artificial Intelligence* (Menlo Park, CA, USA, 2002), American Association for Artificial Intelligence, pp. 392–397.

[14] Coste-Marquis, S., Lang, J., Liberatore, P., and Marquis, P. Expressive power and succinctness of propositional languages for preference representation. In *KR' 04: Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning* (Whistler, Canada, 2004).

[15] Dubois, D., Fargier, H., Perny, P., and Prade, H. On the limitation of ordinal approaches to decision making. In *KR' 02: Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning* (Toulouse, France, 2002), pp. 133–146.

[16] Fargier, H., Lang, J., and Marquis, P. Propositional logic and one-stage decision making. In *KR' 00: Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning* (Breckenridge, CO, USA, 2000), pp. 445–456.

[17] Gaertner, W. *A Primer in Social Choice Theory.* Oxford University Press, Oxford University Press Inc., NY, USA, 2006.

[18] Grabisch, M. On preference representation on an ordinal scale. In *ECSQARU' 01: Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (London, UK, 2001), Springer-Verlag, pp. 18–28.

[19] Hudson, B., and Sandholm, T. Effectiveness of preference elicitation in combinatorial auctions. In *AAMAS '02: Revised Papers from the Workshop on Agent Mediated Electronic Commerce on Agent-Mediated Electronic Commerce IV, Designing Mechanisms and Systems* (London, UK, 2002), Springer-Verlag, pp. 69–86.

[20] Johnson, P. *Social Choice: Theory and Research.* No. 123 in Quantitative Applications in the Social Sciences. Sage Publications, Inc, Thousand Oaks, CA: Sage, 1998.

[21] Konczak, K., and Lang, J. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI'05 (International Joint Conferences on Artificial Intelligence) Workshop on Advances in Preference Handling* (2005).

[22] Lafage, C., and Lang, J. Logical representation of preferences for group decision making. In *KR' 00: Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning* (Breckenridge, CO, USA, 2000), pp. 457–468.

[23] Lafage, C., and Lang, J. Propositional distances and preference representation. In *ECSQARU '01: Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (London, UK, 2001), Springer-Verlag, pp. 48–59.

[24] Lahaie, S., and Parkes, D. Applying learning algorithms to preference elicitation. In *EC' 04: Proceedings of the 5th ACM Conference on Electronic Commerce* (New York, NY, USA, 2004), ACM Press, pp. 180–188.

[25] Lang, J. Some representation and computational issues in social choice. In *ECSQARU' 05: Proceedings of the 8th European Conference Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (Barcelona, Spain, 2005), vol. 3571 of *Lecture Notes in Computer Science*, Springer Berlin/Heidelberg, pp. 15–26. Invited Papers.

[26] Laslier, J.-F. *Tournament Solutions and Majority Voting.* Springer Verlag, Berlin, 1997.

[27] Pini, M., Rossi, F., Venable, K., and Walsh, T. Strategic voting when aggregating partially ordered preferences. In *AAMAS '06: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems* (New York, NY, USA, 2006), ACM Press, pp. 685–687.

[28] Pini, M., Rossi, F., Venable, K., and Walsh, T. Incompleteness and incomparability in preference aggregation. In *IJCAI' 07: Proceedings of the 20th International Joint Conferences on Artificial Intelligence* (2007), pp. 1464–1469.

[29] Pini, M., Rossi, F., Venable, K. B., and Walsh, T. Aggregating partially ordered preferences: impossibility and possibility results. In *TARK '05: Proceedings of the 10th Conference on Theoretical Aspects of Rationality and Knowledge* (Singapore, Singapore, 2005), National University of Singapore, pp. 193–206.

[30] Rossi, F., Venable, K. B., and Walsh, T. Aggregating preferences cannot be fair. *Intelligenza Artificiale 2*, 1 (2005), 30–38.

[31] Tan, S.-W., and Pearl, J. Specification and evaluation of preferences under uncertainty. In *KR' 94: Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning* (San Francisco, CA, USA, 1994), J. Doyle, E. Sandewall, and P. Torasso, Eds., Kaufmann, pp. 530–539.

[32] Wellman, M., and Doyle, J. Preferential semantics for goals. In *AAAI' 91: Proceedings of the 9th National Conference on Artificial Intelligence* (Anaheim, CA, USA, 1991), vol. 2, AAAI Press/MIT Press, pp. 698–703.

[33] Zinkevich, M., Blum, A., and Sandholm, T. On polynomial-time preference elicitation with value queries. In *EC' 03: Proceedings of the 4th ACM Conference on Electronic Commerce* (New York, NY, USA, 2003), ACM Press, pp. 176–185.

Chapter 4

Complexity-Theoretic Aspects of Voting Systems

Both this paper and Bartholdi, Tovey, and Trick (1989b) have been influenced by Nurmi (1984, p. 255), who suggested "... constructing a hierarchy reflecting the difficulty of benefiting from strategic behavior."

Bartholdi, Tovey, and Trick[1]

4.0 Prologue

This chapter surveys complexity-theoretic studies of voting systems. The chapter covers six major problems. These are the problems of finidng winners, electoral control, manipulation in elections, bribery, vote elicitation, and the complexity of communication in voting rules. To facililate complexity-theoretic discussions of voting systems, the chapter begins with an introduction to *Complexity Thoery*. The following sections are dedicated each to one of the following problems in order: *The Complexity of Determining Winners*, *The Complexity of Electoral Control*, *The Complexity of Manipulating Elections*, *The Complexity of Bribery in Elections*, *Complexity of Vote Elicitation (Revisited)*, and *The Complexity of Communication in Voting Rules*. A wealth of literature on the first four problems is available. Hence, *Comments and Bibliographic Notes* appear at the end of the sections corresponding to these problems. The chapter concludes with *New Criteria for Evaluating Voting Rules* in addition to those presented in Chapter 1.

4.1 Complexity Theory: an Introduction

This section briefly reviews some notions from complexity theory. These notions will be frequently used in later discussions. The concepts are briefly presented here in simple terms. For rigorous definitions and full explanation, the reader is referred to standard references

[1]Bartholdi, J., Tovey, C., and Trick, M. (1992) "How hard is it to control an election?" *Mathematical and Computer Modelling (Special Issue on Formal Theories of Politics)* ***16***(8/9):27–40.

on formal languages, algorithms, and complexity theory (such as [41, 45, 47], [20] and [39]). A reader who is familiar with these concepts may skip to the next section.

4.1.1 Problems as Languages

In complexity theory, problem solving can be viewed as language recognition[2]. A problem is stated in the form of a question and has some parameter(s). For example, the problem "what are the dimensions of a rectangle with maximum possible area subject to perimeter length of 24 units?" has the perimeter as a parameter and the entities in question are the dimensions of a maximum area rectangle. Varying the parameter creates a new *instance* of the problem. You can think of the general problem as a template where parameters are indicated by symbols (for the sake of description) and of an instance as replacing the symbolic parameters with actual values. This form of a problem statement where the question requires an optimal value to be computed is called an *optimization problem*. If a problem is stated as a yes or no question, for example, "given an integer n, is n divisible by 3?," then we get a *decision problem*. The question statement of an optimization problem asks for an optimal (best) solution to the problem whilst the question statement of a decision problem asks whether a solution with specific characteristics exists or not. The following is an example showing an optimization problem and the corresponding decision problem:

Definition 4.1.1 (Knapsack (optimization))**.** ***Given*** *a finite set U, a weight $w(u) \in \mathbb{Z}^+$ and a profit $p(u) \in \mathbb{Z}^+$ for each $u \in U$, and a capacity constraint $M \in \mathbb{Z}^+$.*
Find: *A subset $U' \subseteq U$ such that $\sum_{u \in U'} p(u) = P$ is maximized, subject to $\sum_{u \in U'} w(u) \leq M$.*

Definition 4.1.2 (Knapsack (decision))**.** ***Given*** *a finite set U, a weight $w(u) \in \mathbb{Z}^+$ and a profit $p(u) \in \mathbb{Z}^+$ for each $u \in U$, a capacity constraint $M \in \mathbb{Z}^+$, and a target profit $P \in \mathbb{Z}^+$.*
Question: *Is there a subset $U' \subseteq U$ such that $\sum_{u \in U'} w(u) \leq M$ and $\sum_{u \in U'} p(u) \geq P$.*

The answer to the question stated in a decision problem can be determined to be either "yes" or "no" (if the problem is not decidable, we may not be able to find out the answer regardless of the time or space resources available). The instances corresponding to "yes" answers (yes-instances) of a problem constitute a language and the no-instances also constitute a language. The set of all instances also constitute a language. These sets of instances constitute languages in the (limited) sense that a language is a collection of words. Here every instance is a word in that language, and words are composed of the language alphabet and are morphologically structured in a specific way according to the rules of the language. We will not go into this level of detail but will generally assume that these instances are words that are generated systematically from a known alphabet. For more details on the formal definition of languages, see for example [41], [45] or [47].

Problems have to be formalized in order to be tackled systematically by computational algorithms. Therefore, all the problems that we discuss in this chapter are first formally

[2]Complexity theory also studies function problems. For the purpose of the following discussions, we will only focus on language recognition problems in this introduction.

defined where the given instance and question are stated clearly. This applies to all problems discussed from the complexity-theoretic perspective. The framework is characterized by a **Given** part and a **Question** part. This parallels the **input** and **processing** parts of an algorithm, subsequently allowing a computer algorithm solution and ultimately an evaluation of the computational complexity of the problem. These *algorithmic* problems are then classified according to their computational complexity where problems having the same computational complexity (typically defined in terms of space and time resources required to solve the problem) are said to belong to the same complexity class. We next define in more rigorous terms the computational complexity of a problem and briefly mention some important complexity classes.

4.1.2 The Computational Complexity of a Problem

The running time of an algorithm is defined as the number of steps that the algorithm requires to solve the problem. Basic operations such as single addition, subtraction or multiplication are considered to take unit time. The notion of algorithm runtime adopted in complexity theory does not regard the type of machine[3] used or time elapsed in usual units of minutes and seconds, the assumption here is that the runtime depends only on the algorithm and the input. This makes a robust normalized framework for studying the computational complexity of a problem.

So the runtime of an algorithm is a function of the length of the input that depends on the number of steps in the algorithm. For example, if the input is a number x of length n and the number of steps is quadratic in the input size then the algorithm's running time is n^2. If the algorithm's runtime is always less than quadratic then we say that it is $O(n^2)$. In general, for a given function $f(n)$, $O(f(n))$ stands for the set of functions $g(n)$ such that there exist positive constants c and n_0, where $0 \leq g(n) \leq cf(n)$ for all $n \geq n_0$, i.e., the O-notation gives an upper bound. Similarly, if the running time of an algorithm is always more than quadratic in the size of the input (e.g., cubic or exponential), then we say it is $\Omega(n^2)$, so the Ω notation stands for lower bound. If both upper and lower bounds are the same then we obtain a tight (or exact) bound denoted by Θ. The followings are formal definitions of the O-notation, Ω-notation, and Θ-notation in order.

Definition 4.1.3. *For a given function $f(n)$, $O(f(n))$ (pronounced "big-oh of f of n") denotes the set of functions*
$O(f(n)) = \{g(n) :$ *there exist positive constants c and n_0 such that* $0 \leq g(n) \leq cf(n)$ *for all* $n \geq n_0\}$.

Definition 4.1.4. *For a given function $f(n)$, $\Omega(f(n))$ (pronounced "omega of f of n") denotes the set of functions*
$\Omega(f(n)) = \{g(n) :$ *there exist positive constants c and n_0 such that* $0 \leq cf(n) \leq g(n)$ *for all* $n \geq n_0\}$.

[3]Here, by "type of machine" we mean the actual computing platform, i.e., the combination of hardware and software system (e.g., an Intel 80486 processor running DOS Version 6.0), not the theoretical model of computation (e.g., a non-deterministic Turing machine).

Definition 4.1.5. *For a given function $f(n)$, $\Theta(f(n))$ (pronounced "theta of f of n") denotes the set of functions*
$\Theta(f(n)) = \{g(n) :$ *there exist positive constants* c_1, c_2 *and* n_0 *such that* $0 \leq c_1 f(n) \leq g(n)$ *and* $g(n) \leq c_2 f(n)$ *for all* $n \geq n_0\}$.

As with limits in mathematics, these notations are asymptotic, in other words, what they tell us is how the runtime of the algorithm increases as the input size increases. Therefore, only the most dominant summand in the runtime function is considered. For example, if the runtime of some algorithm is $O(n^2 + n)$ then we simply say it is $O(n^2)$, since as n increases, the value of n^2 becomes much more significant than n.

Definition 4.1.6. *The computational complexity of a problem is bounded above by the (asymptotic) worst case running time of the algorithms that solve the problem. If every algorithm that solves the problem requires a specific worst case runtime, then we also obtain a lower bound of the complexity of the problem. In case the upper and lower bounds are asymptotically the same, then we obtain the tight bound or exact complexity of the problem.*

The above assumes that the complexity of an algorithmic problem is usually measured in terms of time requirement. The space dimension is also considered and we have space-complexity as well. Since time is a more critical computational resource, the complexity of a problem is usually measures in terms of time requirements.

4.1.3 Some Important Complexity Classes

P stands for "polynomial" time and it is the complexity class containing problems (or languages) that can be solved (recognized) in polynomial time (by a deterministic Turing machine).

Definition 4.1.7. *A problem belongs to the complexity class* P *of polynomially-solvable problems if it can be solved by an algorithm with polynomial worst-case runtime.*

Recall that a polynomial is a mathematical expression that is the sum of monomials, where each monomial is a product of a coefficient and (positive whole number) power of one or more variables. For example, $x^3 - 7x^2 + 1$.

Generally speaking, this complexity class contains problems that can be efficiently solved, or computationally *tractable.*

NP stands for non-deterministic polynomial time. This is a very important and extensively studied complexity class. In simple terms, the same computational model for studying P is the underlying model for NP as well, except that for NP the next step, while processing the input, is chosen *nondeterministically* from a set of steps. Nondeterminism can be interpreted in terms of parallel computation paths, randomization or guessing[4].

A problem in NP is characterized by polynomial time verification. This means if a solution is given to the problem, its correctness can be verified in polynomial time (in the

[4] The concept of nondeterminism is a big topic, it is usually confusing at first since we tend to conceive computer processing as a deterministic well-defined set of instructions. For introductory interpretations of nondeterminism the reader is referred to [63].

length of the input). Note that all problems in P are also in NP since if a problem can be solved in polynomial time then the solution can also be verified in polynomial time. However, NP also contains problems that can be verified in polynomial time but no polynomial time algorithm was found to solve any of them yet. We still do not know whether this is because those problems *cannot* be solved in polynomial time or because individual research endeavors could not find polynomial time algorithms to solve them. However, since many researchers for many years have independently tried to tackle different problems of this type and no one could solve any of those problems, this is taken as a strong evidence that P is not equal to NP, in other words, there are problem in NP but outside P and they cannot be solved in deterministic polynomial time. The only deterministic algorithms known to solve these problems run in exponential time, other algorithms are based on heuristics or are approximation algorithms.

NP-complete problems are considered to be the hardest problems in NP. An NP-complete problem is at least as hard as any other problem in NP (hence, the term "complete"). A proof that a problem is NP-complete (or NP-hard) is a strong indication that it is not in P and therefore the problem is considered computationally difficult to solve.

#P (read: "sharp P" or "number P") is the complexity class of counting problems associated with decision problems in the class NP. Unlike the other complexity classes listed here, #P is not a class of decision problems but a class of function problems. #P contains all counting problems #A for which there is a polynomial-time non-deterministic Turing machine that for each instance x has exactly as many accepting paths as there are solutions to x. Equivalently, #P is the class of function problems that compute $f(x)$, where f is the number of accepting paths of an NP machine recognizing x. Recall that in decision problems, the question is whether a solution with specific characteristics exits or not, in counting problems the question is *how many* solutions with specific characteristics exist.

$\mathbf{P}_{\parallel}^{\mathbf{NP}}$ is the complexity class containing problems that can be decided in polynomial time via parallel access to NP. The class P^{NP} contains those languages that can be accepted by a deterministic polynomial-time Turing machine with the help of an *oracle* in NP. We can think of an oracle as a subprogram that we call to get an answer for a given query and a call[5] to which is assigned a unit cost. In general, for classes $\mathcal{C}, \mathcal{D}$, the class $\mathcal{C}^{\mathcal{D}}$ contains languages recognized by $\mathcal{C}$ given access to the power of $\mathcal{D}$. Formally, this is defined as $\mathcal{C}^{\mathcal{D}} = \bigcup_{A \in \mathcal{D}} \mathcal{C}^{A}$. We obtain the class $\mathrm{P}_{\parallel}^{\mathrm{NP}}$ if we require the deterministic polynomial-time Turing machine to make all queries to the NP oracle in *parallel*. The following is another definition of $\mathrm{P}_{\parallel}^{\mathrm{NP}}$ in terms of polynomial time truth table reducibility (which is defined in "Operations on Problems and Complexity Classes" below).

Definition 4.1.8. $\mathrm{P}_{\parallel}^{\mathrm{NP}} = \bigcup_{A \in \mathrm{NP}} \{L | L \leq_{tt}^{p} A\}$.

[5] This call is usually termed a *query*.

4.1.4 Operations on Problems and Complexity Classes

Since complexity classes are sets, all set operations such as union, intersection, difference and complement apply to problems and complexity classes. In addition to the basic set operations, we will mention two other types of operations: the disjoint union operation and reductions.

Disjoint union stands for a type of union of two sets that retains the original membership information. To explain this, note that in the (ordinary) union operation if the same element exists in two sets then it appears (once) in the union. In the disjoint union, the elements of each set are marked such that under union of the sets, an element that appears in both sets appears twice as two distinct elements in the union, where each one of these copies is marked in a way indicating its original set. This is why this type of union is also called *marked* union. For simplicity, it is also referred to as the "join" operation. The following is an example. Let $A = \{1, 2, 3\}$ and $B = \{1, 2, 3, 4\}$. Then the disjoint union of A and B, denoted[6] $A \cup^* B$, is found by first marking the elements of each set as follows:
$A^* = \{(1, 0), (2, 0), (3, 0)\}$, $B^* = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$ or any other marking that makes it possible to distinguish between element a that comes from set A and element a that comes from set B. The disjoint union is then simply the union of A^* and B^* which is $\{(1, 0), (2, 0), (3, 0), (1, 1), (2, 1), (3, 1), (4, 1)\}$.

Reductions are of many types. The basic idea is that some problems can be *reduced* to other problems. If we know the solution to the problem we are *reducing to* then we can solve the problem we are reducing (from). Conversely, if we know that the problem we are reducing is difficult then we conclude that the problem we are reducing to is also difficult. For simplicity, one can think of "problem A is reducible to B" as "problem A is *less than or equal in difficulty* than B." Out of the various types of reductions, polynomial time reduction is widely known and used. The followings are formal definitions of reduction types that will be referred to later in this chapter.

Definition 4.1.9 (Polynomial-Time Many-One Reducibility)**.** *If A and B are two problems, then A is polynomial-time many-one reducible to B, denoted $A \leq_m^p B$, if there is a function f that maps every instance of A to an instance of B such that the first instance (of A) is a yes-instance if and only if the second instance (of B) is a yes-instance[7], and f can be computed in polynomial time.*

Polynomial time truth table reducibility is another type of reductions and is defined as follows:

Definition 4.1.10 (Polynomial-Time Truth Table Reducibility, $\leq_{tt}^p$)**.** *$A \leq_{tt}^p B$ if and only if there is a polynomial-time machine M such that $L(M^B) = A$ and M asks all its questions to B in parallel and receives all their answers simultaneously.*

[6]Sometimes, the disjoint union is denoted by $A \oplus B$ but since this symbol is also used to stand for the symmetric difference of two sets (which is the union minus intersection), it is preferred to use $\cup^*$ to stand for the disjoint union.

[7]In other words, both are yes-instances or both are no-instances.

4.1.5 The Notion of Completeness

Complete problems for any complexity class are the hardest problems in that class. That is, a complete problem for any complexity class is considered to be at least as hard as any problem in the class and a solution to a complete problem implies solutions to all problems in that class. The following is a formal definition of completeness (polynomial time reduction ($\leq_m^p$) is defined in "Operations on Problems and Complexity Classes" above).

Definition 4.1.11. *$\mathcal{C}$-hard For any class $\mathcal{C}$, we say that set A is $\mathcal{C}$-hard if and only if for all $B \in \mathcal{C}$, $B \leq_m^p A$.*

Definition 4.1.12. *$\mathcal{C}$-complete For any class $\mathcal{C}$, we say that set A is $\mathcal{C}$-complete if and only if $A \in \mathcal{C}$ and A is $\mathcal{C}$-hard.*

To prove the NP-completeness of a problem, it is sufficient to show that the problem belongs to NP (which is usually shown easily) and then to reduce an NP-complete problem to it.

Some NP-complete Problems

The following is a list of NP-complete problems. These are the problems mostly used in complexity analysis of voting and related problems. These problems are all stated as in [31].

Definition 4.1.13 (Partition)**.** ***Given*** *a finite set A and a "size" $s(a) \in \mathbb{Z}^+$ for each $a \in A$.*
Question: *Is there a subset $A' \subseteq A$ such that $\sum\limits_{a \in A'} s(a) = \sum\limits_{a \in A - A'} s(a)$?*

Definition 4.1.14 (Knapsack (decision))**.** ***Given*** *a finite set U, a weight $w(u) \in \mathbb{Z}^+$ and a profit $p(u) \in \mathbb{Z}^+$ for each $u \in U$, a capacity constraint $M \in \mathbb{Z}^+$, and a target profit $P \in \mathbb{Z}^+$.*
Question: *Is there a subset $U' \subseteq U$ such that $\sum\limits_{u \in U'} w(u) \leq M$ and $\sum\limits_{u \in U'} p(u) \geq P$?*

Definition 4.1.15 (Exact Cover By 3-Sets (X3C))**.** ***Given*** *a finite set X of size $3q$ and a collection C of r subsets of X (where $r > q$) each of size 3.*
Question: *Does C contain an exact cover for X, that is, a subcollection $C' \in C$ such that every element of X occurs in exactly one member of C'?*

Definition 4.1.16 (Cover By 3-Sets (3-Cover))**.** ***Given*** *a finite set X of size $3q$ and a collection C of subsets $S_{i_{1 \leq i \leq r}}$ of X (where $r > q$) each of size 3.*
Question: *Is there a cover of X consisting of q of the subsets?*

Definition 4.1.17 (Feedback Arc Set)**.** ***Given*** *a directed graph $G(V, E)$ and a positive integer $k \leq \|E\|$.*
Question: *Is there a subset of no more than k arcs (directed edges) which includes at least one arc (directed edge) from every cycle in G?*

4.1.6 The Polynomial Hierarchy

Different complexity classes have different properties and different logical descriptions of problems they contain. This description suggests that these classes form a hierarchy of classes that are thought to be distinct[8].
The polynomial hierarchy is formally defined as follows:

Definition 4.1.18 (Polynomial Hierarchy). *Let* $\Sigma_0 = \mathrm{P}$*,* $\Sigma_1 = \mathrm{NP}$*, and* $\Sigma_{k+1} = \mathrm{NP}^{\Sigma_k}$*. The polynomial hierarchy (PH) is the union of all* Σ_k *for* $k \geq 1$ *.*

We will see next how this hierarchal structure of complexity classes inspired the early work on complexity-theoretic analysis of voting schemes.

4.1.7 Complexity Theory and the Study of Voting

The computational complexity of voting schemes and related problems was initially studied by Bartholdi, Tovey, Trick, and Orlin. In a series of papers that appeared in the late eighties-early nineties, these pioneers of what is now known as computational social choice presented a complexity theoretic treatment of some voting schemes and studied the computational complexity of two related problems: manipulation and control [5, 4, 3, 6]. In "How hard is it to control an election?," Bartholdi, Tovey and Trick note how the classification of voting systems according to their computational complexity was inspired by Nurmi's comment on the hierarchical difficulty of different levels of voters' strategic behavior:

> "The classification of voting procedures by their computational complexity has also appeared in Bartholdi, Narasimhan, and Tovey (1990), Bartholdi, Tovey, and Trick (1989a), and Bartholdi, Tovey, and Trick (1989b). Both this paper and Bartholdi, Tovey, and Trick (1989b) have been influenced by Nurmi (1984, p. 255), who suggested '... constructing a hierarchy reflecting the difficulty of benefiting from strategic behavior' " [6].

In the following sections of this chapter, we will discuss the main problems in voting that are studied from the complexity theory perspective. These are in order: the complexity of determining winners, the complexity of control, the complexity of manipulation and strategic voting, the complexity of bribery, the complexity of vote elicitation, and the complexity of communication in voting rules.

[8] These classes are thought to be distinct with respect to which sets each includes, i.e., with respect to how many more problems each higher class includes or how much more computational power each higher class has.

4.2 The Complexity of Determining Winners

4.2.1 Background

Perhaps one of the earliest problems studied in what is now called computational social choice is the computational complexity of determining winners. The complexity of determining the winner set and the complexity of the voting system are used interchangeably in the literature. This reflects the fundamental importance of this problem in the computational study of voting systems. Moreover, using the general description "complexity of a voting system" to refer to the complexity of determining winners (under that system) is apt. After all, the main goal is to declare winners, based on the votes. This is the function of a voting system, and hence its computational complexity is the computational complexity of determining winners.

Most widely used voting rules, such as Plurality, have a low complexity which is a necessity for the efficient operation of any voting system (especially if paper ballots are used which is the case with large scale national presidential elections). However, the investigation of voting rules' computational complexity, began with the discussing "voting schemes for which it can be difficult to tell who won the election" [5]. About a decade before that paper appeared, the empricial difficulty of these schemes was noted by Fishburn [30]. This difficulty was first formally presented in Bartholdi, Tovey, and Trick's work.

In addition to this novel exposition of two voting schemes with high complexity, their work presented a new perspective and new tools for studying and evaluating voting rules: computational complexity theory.

In this section, we present a range of related work and study computational aspects of the winner problem.

4.2.2 Definitions

The winner problem is formally defined as follows:

Definition 4.2.1 ($\mathcal{E}$-Winner)**.** *Given a set of candidates C, a distinguished candidate $c \in C$, and a set of voters V specified by their preference orders*[9] *on C, and a voting rule $\mathcal{E}$.*
Question: *Is c a winner under $\mathcal{E}$?*

Definition 4.2.2 ($\mathcal{E}$-Ranking)**.** *Given a set of candidates C, distinguished candidates $c, c' \in C$, and a set of voters V specified by their preference orders on C.*
Question: *Does c defeat c' in the $\mathcal{E}$-election?*

We will refer to this definition in the following discussion of the complexity of determing winners.

4.2.3 The Beginning: Difficult Voting Schemes

The start of complexity theoretic study of voting systems as a field is marked with the early work of Bartholdi, Tovey, and Trick [5] which appeared in the late eighties. They

[9] In other words, their preferences are strict (irreflexive and antisymmetric), transitive, and complete.

presented three well-defined voting schemes and showed that although these schemes have nice properties, they are computationally difficult to compute[10].

Next are the formulations of these problems and main results regarding their computational complexity.

Definition 4.2.3 (Dodgson Score)**.** *Given a set of candidates C, a distinguished candidate $c \in C$, and a set of voters V specified by their preference orders on C, and a positive integer k.*
Question: *Is the Dodgson score of c less than or equal to k?*

It was proven in [5] that this problem is NP-complete. Showing membership in NP is simple since we can verify a "yes" answer to the previous question in polynomial time by identifying appropriate switches and counting the votes. NP-hardness was proven via a reduction the (NP-complete problem) X3S (Exact Cover by 3 Sets).

Theorem 4.2.1. Dodgson-Score *is* NP-*complete [5].*

A similar approach was used to obtain the following results:

Theorem 4.2.2. Dodgson-Ranking *is* NP-*hard [5].*

Theorem 4.2.3. Dodgson-Winner *is* NP-*hard [5].*

Remark: This result shows only NP-hardness. Recall that NP-hard means NP-complete *or* harder, so the question of whether Dodgson-Ranking is NP-complete was left open in this work.

Although the general complexity for Dodgson-Winner is high, and so it is a computationally difficult problem, it was pointed out that when either the candidate set or the vote set is bounded by a constant, then the problem becomes solvable in polynomial time using integer linear programming [5]. The authors further noted that "the effort required to determine a Dodgson winner appears to increase more quickly as a function of $|C|$ than as a function of $|V|$."

It was also shown by Rothe et al. that a homogenous variant of Dodgson's scheme can be solved efficiently also by linear programming [58].

There is yet more progress in this direction. When the number of voters is much greater than the number of candidates (which is the case in many voting applications), a polynomial time greedy algorithm designed by Homan and Hemaspaandra was proven to very frequently find Dodgson winners. The algorithm moreover knows that it found the winners (correctly) and it never declares a nonwinner to be a winner [40].

Similar general results were obtained for Kemeny's voting rule. NP-hardness was shown via a reduction from Feedback-Arc-Set. They are listed next.

Theorem 4.2.4. Kemeny-Score *is* NP-*complete [5].*

[10]These voting systems and related terms are defined in Chapter 1 of this thesis.

Remark: This result was independently obtained by Orlin[11] and by Wakabayashi [62]. A similar approach was used to obtain the following results:

Theorem 4.2.5. Kemeny-Ranking *is* NP*-hard [5].*

Theorem 4.2.6. Kemeny-Winner *is* NP*-hard [5].*

Since the Kemeny rule is computationally hard (even when there are only four input rankings to aggregate [24]), researchers studied polynomial time approximations for this rule. The first 2-approximation (i.e., the approximation differs from the optimal solution by a factor of 2) algorithms were given in [22, 24]. An improved 11/7-approximation is given in [1][12].

In concert with Arrow's impossibility theorem, Bartholdi, Tovey, and Trick suggested a new model of theorems that they called "impracticality" theorems. The first was a theorem that says basically that any voting system satisfying a specific set of criteria is impractical (recall that Arrow's theorem basically says that any social welfare function satisfying a specific set of criteria is impossible). In Chapter 1, when defining Kemeny's voting rule, we mentioned that it is the only voting rule that satisfies neutrality, consistency and the Condorcet criterion. Hence, this was the impracticality theorem proven by Bartholdi et al. [5]:

Theorem 4.2.7 (Impracticality Theorem)**.** *Under any voting scheme that is neutral, consistent, and Condorcet, the winner problem is* NP*-hard.*

Just before the conclusion of their groundbreaking paper, Bartholdi et al. posed the question of whether there are stronger versions of this theorem. Was this question answered? We will see next.

4.2.4 The Continuation: More-Difficult Voting Schemes

As noted previously, Bartholdi et al. left open the question of whether Dodgson and Kemeny schemes are NP-complete. For Dodgson's scheme, this question was answered by Hemaspaandra, Hemaspaandra, and Rothe [35] and for emeny's scheme, it was answered later by Hemaspaandra, Spakowski, and Vogel [38]. The answer in both cases was "No." Moreover, in each of these cases it turned out that the exact complexity of these systems is higher than NP-complete. It was shown that both problem were $\mathrm{P}_{\|}^{\mathrm{NP}}$-complete (or equivalently Θ_2^p-complete), i.e., they belong to the class of languages that are complete for parallel access to NP[13]. This was obtained by improving on the previous lower bound (NP-hard) and then by providing matching upper bounds. Combining both bounds, the exact complexity of these schemes was obtained. The same result was also obtained for oung's election systems [58] . The three following theorems are listed in chronological order.

[11] It is mentioned in [5] that Orlin established the NP-completeness of Kemeny-Score by private correspondance.

[12] In [1], computing a Kemeny ranking is referred to as the Rank-Aggregation problem.

[13] Parallel access to NP and the notion of completeness are defined in section 4.1.

Theorem 4.2.8. Dodgson-Winner *is* $\mathrm{P}_{\|}^{\mathrm{NP}}$*-complete [35].*

Remark: Dodgson-Winner is considered to be the *first natural problem* that is complete for parallel access to NP. Before this result, the class $\mathrm{P}_{\|}^{\mathrm{NP}}$ was empty of "natural" complete problems. Given that voting systems are natural methods for aggregating preferences, the winner problem of a voting scheme that—in addition—dates back to the nineteenth century-was widely accepted as a natural difficult problem in the class $\mathrm{P}_{\|}^{\mathrm{NP}}$.

Theorem 4.2.9. Young-Winner *is* $\mathrm{P}_{\|}^{\mathrm{NP}}$*-complete [58].*

Theorem 4.2.10. Kemeny-Winner*is* $\mathrm{P}_{\|}^{\mathrm{NP}}$*-complete [38].*

We have demonstrated in a previous chapter that these three voting systems have something in common: they are all Condorcet consistent. In particular, they were all proposed to extend the Condorcet rule by respecting Condorcet winners yet overcoming the cyclic majority problem. Nevertheless, the techniques used for obtaining these results are quite different and the three problems were not reduced to each other.

A $\mathrm{P}_{\|}^{\mathrm{NP}}$-complete problem is considered to be extremely difficult in terms of computational complexity, since not only no polynomial time algorithm exists for such problems (unless P = NP) but unless the polynomial hierarchy collapses (to NP), these problem are outside the class NP.

Thereupon, another open question in Bartholdi, Tovey, and Trick's paper was answered and a stronger version of their practicality theorem was proven:

Theorem 4.2.11 (Optimal Impracticality Theorem)**.** *Under any voting scheme that is neutral, consistent, and Condorcet, the winner problem is* $\mathrm{P}_{\|}^{\mathrm{NP}}$*-complete.*

It seems that satisfying a number of criteria comes at the cost of computational feasibility.

4.2.5 Multi-Winner Voting Schemes

In single-winner voting schemes, the assumption is that a vote completely reflects the political opinion of the voter. This assumption is demonstrated mostly in the use of plurality rule. Since this assumption is not realistic, many alternative voting schemes have been suggested to overcome this weakness of simple plurality. Among these alternative schemes are the Borda count, approval voting, Copeland and distinctly antiplurality. In comparison to plurality, these schemes try to realize the goal of *full proportional representation.* In the case of multi-winner schemes, only few alternatives were suggested as fully proportional representative schemes. Among these are two methods: Chamberlin and Courant's scheme [8] and Monroe's scheme [49].

To explain these two schemes, we first need the following definitions. Let u_{ic} quantify[14] the degree to which candidate c *misrepresents* voter i. Assume that the scale of u_{ic} can be normalized over all voters for comparison purposes. For example, for a scale

[14] The details of how this value is calculated are irrelevant here. We only assume that this value is known.

from 0 to $m-1$, where m is the number of candidates, let $u_{ic}=0$ be the misrepresentation value of i's favorite candidates and $u_{ic'}=m-1$ be the misrepresentation value for i's least preferred candidates. For all other candidates this value is between 0 and $m-1$. Let $\mathcal{S}=\{S\subseteq C : \|S\|=k\}$ by the collection of all possible subsets of the set of candidates of size k. Let $f_S : V \rightarrow S$ be a function that assigns voters to candidates based on misrepresentation values. The misrepresentation value of a voter i under this assignment is $u_{i\,f_S(i)}$. The total misrepresentation of f_S is $\Sigma_{i\in V} u_{i\,f_S(i)}$. The goal of Chamberlin and Courant's scheme is to find a subset of k candidates that minimize that misrepresentation. The subset of k candidates that optimally minimizes representation for all voters, is the winner set. Chamberlin and Courant use weighted voting to attain even proportionality. Monroe's scheme has a different approach for achieving balanced proportionality. Monroe's scheme is similar to Chamberlin and Courant's but has one restriction; in the voters-candidate assignment, voters are evenly distributed among candidates in every subset of k candidates. Hence, each candidate in the winner set is supported by roughly the same number (n/k) of voters[15]. The details of how an even distribution can be achieved are given in [49].

So the problem in these schemes is basically an optimization problem (minimizing misrepresentation).

Consider the following definition of deciding whether a set of candidates has misrepresentation value that does not exceed a certain limit:

Definition 4.2.4. *Given a set of n voters V, a set of candidates C, the number of winners $k\in\mathbb{N}$, misrepresentation values*[16] *$u_{ic}\in\{0,1,\ldots,m\}$ for $1\leq i\leq n$, and $t\in\mathbb{N}$. We are asked whether there exists a subset $S\subseteq C$ such that $\|S\|=k$, with misrepresentation at most k [56].*

Clearly, the problem of determining winners in the abovementioned (multi-winners) schemes is harder than the problem just defined. The following related theorems are due to Procaccia et al. [56]:

Theorem 4.2.12. *Chamberlin and Courant's rule-Winners is* NP*-complete.*

Theorem 4.2.13. *Monroe's rule-Winners is* NP*-complete.*

Theorem 4.2.14. *When the number of winners k satisfies $k=O(1)$ (i.e., is a constant) then Chamberlin and Courant's rule-Winners is in* P*.*

Theorem 4.2.15. *When the number of winners k satisfies $k=O(1)$ then Monroe's rule-Winners is in* P*.*

Results show again that satisfying desirable criteria (such as full proportional representation) comes at the cost of computational feasibility.

[15] When deriving complexity results, n/k is assumed to be an integer. This assumption does not compromise the generality of the result as pointed in [56] since the set of voters can be padded with enough number of voters with $u_{ic}=0$ for all $c\in C$ such that n/k is an integer.

[16] In the definition given in [56], misrepresentation values range between 0 and m inclusively.

4.2.6 Determining Winners as Graph Problems

Sequential Majority Voting

Recall the binary voting rule defined in Chapter 1, the majority graph G defined in Chapter 3, and the set of possible winners defined in Chapter 3. A top cycle in the majority graph corresponds to a set of k alternatives c_i, were $k > 2$ and $1 \leq i \leq k$ such that c_1 beats c_2, c_2 beats c_3, ..., c_{k-1} beats c_k and c_k beats c_1, and every alternative not in the top cycle is beaten by some alternative in the top cycle.
The following is a characterization of possible winners in terms of paths in G.

Theorem 4.2.16. *Given a complete majority graph $G(C, E)$, a candidate c is a possible winner if and only if, for every other candidate c' in C, there exists a path from c to c' [50].*

Based on this result and the fact that path finding in graphs is in P, the following result was obtained:

Theorem 4.2.17. *Given a complete majority graph $G(C, E)$ and a candidate c, checking whether c is a possible winner and if so finding a tree where c wins is in* P *[44].*

The previous results apply to binary voting trees in general, however, since not all binary trees are balanced trees, a binary voting tree might correspond to an election where some candidate (possibly the winner) competes with very few other candidates. Hence, yielding an unfair competition (especially if this candidate ends up being the winner). A competition is considered fair if the corresponding binary tree is balanced[17]. A fair possible winner is a winner in some balanced binary tree. Since the Condorcet winner is the winner of all possible binary trees corresponding to pairwise sequential majority elections, the Condorcet winner, in the context of graph representations of elections, is the best depiction of a fair winner. Unfortunately, the complexity of determining fair possible winners is not known yet, moreover, in the only setting for which it is known—weighted voting—it is computationally difficult to check whether a given candidate is a *fair* possible winner.

Theorem 4.2.18. *Given a complete majority graph $G(C, E)$ and a candidate c, it is* NP-*complete to check whether c is a fair possible winner for G [44].*

Kemeny Rankings

Recall the Kemeny rule defined in Chapter 1, and discussed earlier in this section. The Kemeny voting rule can be thought of as an optimization problem where the goal is to produce a ranking of all alternatives that *minimizes* the number of disagreements with n individual rankings, or equivalently *maximizes* the number of agreements with the individual rankings.

Given the majority graph, we can assign weights to G and reinterpret Kemeny's voting rule as a graph problem. Consider the majority graph G, where vertices represent candidates and for any two candidates $a, b \in C$, let h_{ab} be the number of voters who prefer a to

[17] This implies the assumption that the number of candidates is a power of two, however, this assumption does not restrict the generality of the problem.

b and h_{ba} be the number of voters who prefer b to a. For an edge (a, b) in G, let the weight of this edge be h_{ab} minus h_{ba}. Note that this number is always nonnegative since an edge is drawn from vertex a to vertex b only if more voters prefer a to b.

If an edge (a, b) exists in G and a Kemeny ranking ranks a above b, then the Kemeny ranking agrees with h_{ab} individual rankings on the pair (a, b) otherwise if the Kemeny ranking ranks b above a, then this Kemeny ranking agrees with h_{ba} individual rankings (which is a smaller number of agreements). Given the way we assigned weights to edges, when the direction of an edge does not agree with the Kemeny ranking then the number of disagreements with the Kemeny ranking is equal to the weight of that edge. Hence we can translate the Kemeny rule optimization problem into a graph optimization problem where the goal is to minimize the total weight of edges that disagree with the final ranking.

Initially, a lower bound on the total weight of edges that disagree with the Kemeny ranking was proposed by Davenport and Kalagnanam [21]. This lower bound was obtained by observing that for each cycle in a set of edge-disjoint cycles in G, at least one edge disagrees with the Kemeny ranking (since the Kemeny ranking must be transitive), since the goal is to minimize the total weight, for every cycle in G, the edge of the smallest weight is augmented to the lower bound. Several improved lower bounds were later proposed by Conitzer, Davenport and Kalagnanam [11]. These were obtained by looking at general cycles (not necessarily edge-disjoint) and using linear programs.

4.2.7 Summary of Results

Table 4.1 summarizes results discussed in this section and table 4.2 summarizes results related to the winner problem discussed in this section and in Chapter 3. Some of the latter results will be revisited and discussed in more detail in a later section of this chapter.

Table 4.1: Summary of results on complexity of determining winners sorted by increasing order of complexity.

Voting Rule	**Complexity**	**Remark**	**Result in**
Plurality	P	$O(\lvert C\rvert + \lvert V\rvert)$	[5]
Positional scoring protocols	P		
Dodgson	P	$\lVert C\rVert$ is constant	[5]
Dodgson	P	$\lVert V\rVert$ is constant	[5]
Dodgson homogenous variant	P		[58]
Dodgson	P	greedy heuristic algorithm "frequently correct" when $\lVert V\rVert \gg \lVert C\rVert$	[40]
Chamberlin and Courant's k multi-winner scheme	P	if k is constant	[56]
Monroe's k multi-winner scheme	P	if k is constant	[56]
Chamberlin and Courant's k multi-winner scheme	NP-complete	even when u_{ic} values are in binary	[56]
Monroe's k multi-winner scheme	NP-complete	even when u_{ic} values are in binary	[56]
Sequential majorty weighted voting	NP-complete	fair possible winner	[44]
Kemeny	$\mathrm{P}_{\parallel}^{\mathrm{NP}}$-complete	tight bound	[38]
Dodgson	$\mathrm{P}_{\parallel}^{\mathrm{NP}}$-complete	tight bound	[35]
Young	$\mathrm{P}_{\parallel}^{\mathrm{NP}}$-complete	tight bound	[58]

Table 4.2: Summary of results on finding possible/necessary winners under incomplete preferences.

Problem	**Voting Rule**	**Complexity**	**Result in**
Necessary Winners	Condorcet	P	[42]
Necessary Winners	Positional Scoring Protocols	P	[42]
Necessary Winners	STV	coNP-complete	[53]
Possible Winners	Scoring Protocols	P	[42]
Possible Winners	STV	NP-complete	[53]
Possible Winners	Condorcet	P	[42]
Fair Possible Winners	Weighted sequential majority	NP-complete	[44]
Is $c \in C$ a Necessary Winner?	Any $\mathcal{E} \in$ P	coNP	[42]
Is $c \in C$ a Necessary Winner?	General $\mathcal{E}$	coNP-hard	[53]
Is $c \in C$ a Possible Winner?	Any $\mathcal{E} \in$ P	NP	[42]
Is $c \in C$ a Possible Winner?	General $\mathcal{E}$	NP	[53]
Determining winners via Vote Elicitation	Any $\mathcal{E} \in$ P satisfying IIA	P	[53]
Finding $PW^* \subseteq PW$ s.t. $\|PW^*\| < k\|PW\|$, $k \in \mathbb{Z}^+$	STV	NP-hard	[53]
Manipulation	Any $\mathcal{E}$ s.t. PW, NW $\in$ P	P	[42]

4.2.8 Comments and Bibliographic Notes

- Although the complexity theoretic analysis of voting rules [5] is widely known as the first computational study of voting rules, at least a decade earlier, some work studied the empirical computational complexity of voting rules [30]. Also, the early result of the difficulty in finding Kemeny scores, was proven independently by Orlin[18] and by Wakabayashi [62] few years before that.

- Among the topics discussed in this section was the graph interpretation of some voting rules. A work that has not been discussed here is related to computing Slater rankings (using similarities among candidates and hieratical election graphs) by Conitzer [10]. Another interpretation of voting rules is based on viewing them as "maximum likelihood estimators" of correct outcomes where we view optimal alternatives as correct outcomes and the vote of each voter as a "noisy perception" of this correct outcome. This interpretation is dated back to Condorcet [46] and continued recently by Young [65], and Conitzer and Sandholm [16].

[18] Private correspondance.

4.3 The Complexity of Electoral Control

4.3.1 Electoral Control

Electoral control refers to the election chair's attempt or action to change the outcome of the election by *controlling* its structure. This involves procedural aspects of conducting elections such as the way the election agenda is set such as the sequence in which candidates run against each other or the way voters are partitioned and grouped.

The control problem is a natural one in many voting settings. As was mentioned earlier, these settings occur in a wide range of domains. Therefore, investigating the issue of control, which is closely and naturally related to voting, is central in the study of voting systems. Moreover, since complexity-theory is used as measure through which this problem is studied, the control problem has a special and important place in the study of complexity-theoretic aspects of voting systems.

The work on the complexity of control problems was pioneered by Bartholdi, Tovey, and Trick [6]. In "How hard is it to control an election?," they defined the control problem, and discussed its complexity. The work focused on different types of control and results were shown for the two known voting rules: Plurality and Condorcet. The different types of control and the computational complexity of each of them are discussed next.

4.3.2 Types of Control

Two types of control are studied in the literature of electoral control complexity. These are the constructive control which was studied first [6], and the destructive control which followed in the work of Hemaspaandra, Hemaspaandra and Rothe [36].

There is a different classification for the types of control problems. It is the classification that focuses on whether the control is exercised on the voters and voting related aspects or on the candidate set and related aspects (e.g., tie-breaking rules). The emphasis in a given context depends on the key concepts and on the goal of the discussion. Since this work surveys related research studies, the types of control problems are presented here according to the main theme of each of the related works in chronological order. This seems to serve both the nature and purpose of this work and also makes it easy to understand the development of research on the control problem.

Next are the definitions of the different control problems, formulated as they were first presented by the authors in the corresponding work. The formulation of these problems is consistent with the conventions of defining problems in complexity theory. This thesis too, follows this line throughout. The notation in the formal definitions in this section assume basic knowledge of how elections and related entities are denoted. Chapter 1 discusses notation and basic definitions.

Constructive Control

In the problem of constructive control, we ask the question of whether the authority conducting the election (or the chairman) can adjust some procedural matters so that a given candidate c, who is not a winner of the election if no control is executed, becomes the *unique* winner of the election? What is the cost of achieving this goal if achievable? And how to achieve it if it is feasible? It is assumed that the chairman knows the preferences of the voters in advance.

Constructive control is accomplished in a number of ways. These are: adding candidates, deleting candidates, partition of candidates, run-off partition of candidates and adding voters, deleting voters and partition of voters.

Destructive Control

In the problem of destructive control, we ask the question of whether the chairman can adjust some procedural matters so that a given candidate c, who is the winner of the election if no control is executed, does *not* become the *unique* winner? What is the cost of achieving this goal if achievable and how to achieve it (if it is achievable or it is computationally easy to achieve)?

In this type of control also, it is assumed that the chairman knows the preferences of the voters a priori.

Regarding the terminology, here the term control "type" refers to constructive or destructive control and the term control "method" refers to the procedure through which the desired outcome of the control is realized, namely: adding candidates, deleting candidates, partitioning of candidates, run-off partitioning of candidates and adding voters, deleting voters and partitioning of voters.

The term "model" of control is used here in a way similar to the way it was used in some related work, to stand for the specification of the type of control and the method used *together*, for example "destructive control via adding candidates" is a model of control. However, it is used here to precisely mean that, although it was used loosely in the literature to stand for a method of control or a model of control.

4.3.3 How is Control Exercised?

Constructive or destructive control is accomplished by means of adding candidates, deleting candidates, partition of candidates, run-off partition of candidates and adding voters, deleting voters and partition of voters. Tie-breaking rules, whether tie-promote or tie-eliminate, also play an important role in determining the outcome and sometimes the complexity of the control.

The following are formulations of each of these control methods (given for the constructive and destructive cases).

The following control methods pertain to the candidate set:

Definition 4.3.1 (Control by Adding Candidates)**.** *Given a set C of qualified candidates and a distinguished candidate $c \in C$, a set D of possible spoiler (added) candidates, and a set V of voters with preferences over $C \cup D$.*

- ***Question (constructive)****: Is there a choice of candidates from D whose entry into the election would guarantee that c is the unique winner?*

- ***Question (destructive)****: Is there a choice of candidates from D whose entry into the election would guarantee that c is not the unique winner?*

Definition 4.3.2 (Control by Adding $\leq k$-Candidates)**.** *Given a set C of qualified candidates and a distinguished candidate $c \in C$, a set D of possible spoiler (added) candidates, a positive integer $k \leq \|D\|$, and a set V of voters with preferences over $C \cup D$.*

- ***Question (constructive)****: Is there a choice of at most k candidates from D whose entry into the election would guarantee that c is the unique winner?*
- ***Question (destructive)****: Is there a choice of at most k candidates from D whose entry into the election would guarantee that c is not the unique winner?*

Definition 4.3.3 (Control by Deleting Candidates)**.** *Given a set C of qualified candidates and a distinguished candidate $c \in C$, a set V of voters and a positive integer $k < \|C\|$.*

- ***Question (constructive)****: Is there a set of k or fewer candidates in C whose disqualification would guarantee that c is the unique winner?*
- ***Question (destructive)****: Is there a set of k or fewer candidates in $C - \{c\}$ whose disqualification would assure that c is not the unique winner?*

Note that if we replace $C - \{c\}$ by C in the last question then the answer is trivially: Yes! this is the singleton set $\{c\}$.

Definition 4.3.4 (Control by Partition of Candidates)**.** *Given a set C of qualified candidates and a distinguished candidate $c \in C$ and a set V of voters.*

- ***Question (constructive)****: Is there a partition of C into C_1 and C_2 such that c is the unique winner in the sequential two-stage election in which the winners in the subelection (C_1, V) (after applying the tie-handling rule move forward to face the candidates in C_2 (with voter set V)?*
- ***Question (destructive)****: Is there a partition of C into C_1 and C_2 such that c is not the unique winner in the sequential two-stage election in which the winners in the subelection (C_1, V) who survive the tie-handling rule move forward to face the candidates in C_2 (with voter set V)?*

Definition 4.3.5 (Control by Run-Off Partition of Candidates)**.** *Given a set C of qualified candidates and a distinguished candidate $c \in C$ and a set V of voters.*

- ***Question (constructive)****: Is there a partition of C into C_1 and C_2 such that c is the unique winner of the election in which those candidates winning (with respect to the tie-handling rule) subelections (C_1, V) and (C_2, V) have a run-off with voter set V?*
- ***Question (destructive)****: Is there a partition of C into C_1 and C_2 such that c is not the unique winner of the election in which those candidates winning (with respect to the tie-handling rule) subelections (C_1, V) and (C_2, V) have a run-off with voter set V?*

Definition 4.3.6 (Control by Effective Sequence of Comparisons[19]). *Given a set C of qualified candidates, a distinguished candidate $c \in C$, and a set V of voters.*

- ***Question (constructive)**: Is there a sequence of pairwise comparisons between candidates such that c is the unique winner under the sequential pairwise voting rule?*
- ***Question (destructive)**: Is there a sequence of pairwise comparisons between candidates such that c is not the (unique) winner under the sequential pairwise voting rule?*

The following control methods pertain to the voter set:

Definition 4.3.7 (Control by Adding Voters). *Given a set C of qualified candidates and a distinguished candidate $c \in C$, a set V of registered voters, an additional set V' of yet unregistered voters (both V and V' have preferences over C), and a positive integer $k < \|V'\|$.*

- ***Question (constructive)**: Is there a set of k or fewer voters from V' whose registration would guarantee that c is the unique winner?*
- ***Question (destructive)**: Is there a set of k or fewer voters from V' whose registration would guarantee that c is not the unique winner?*

A generalization of control by adding voters for multi-winner voting systems was presented in [56]. However, this definition uses utility functions to evaluate candidates. Winners in this definition are candidates with maximum utility.

Definition 4.3.8 (Multi-Winner-Control). *Given a set C of candidates, a set V of voters, a set V' of unregistered voters, the number of winners $m \in \mathbb{N}$, a utility function $u : C \to \mathbb{N}$, the number of winners we are allowed to register $k \in \mathbb{N}$, and an integer $t \in \mathbb{N}$.*

- ***Question**: Is it possible to register at most k voters from V' such that in the resulting election, $\Sigma_{c \in W} u(c) \geq t$, where W is the set of winners and $\|W\| = m$.*

Definition 4.3.9 (Control by Deleting Voters). *Given a set C of qualified candidates and a distinguished candidate $c \in C$, a set V of voters and a positive integer $k < \|V\|$.*

- ***Question (constructive)**: Is there a set of k or fewer voters in V whose removal (disenfranchisement) would guarantee that c is the unique winner?*
- ***Question (destructive)**: Is there a set of k or fewer voters in V whose removal (disenfranchisement) would guarantee that c is not the unique winner?*

Definition 4.3.10 (Control by Partition of Voters). *Given a set C of qualified candidates and a distinguished candidate $c \in C$ and a set V of voters.*

- ***Question (constructive)**: Is there a partition of V into V_1 and V_2 such that c is the unique winner in the hierarchical two-stage election in which the winners of (C, V_1) and (C, V_2) run against each other with voter set V?*

[19] This definition applies only to rules where sequential pairwise comparisons take place. This type of agenda control was discussed in previous work (e.g., [2]), however, it is first formalized in the context of computational social choice as one of standard types of control here.

- ***Question (destructive)***: *Is there a partition of V into V_1 and V_2 such that c is not the unique winner in the hierarchical two-stage election in which the winners of (C, V_1) and (C, V_2) run against each other with voter set V?*

4.3.4 Important Notions of Control Complexity

The answers to the above questions (for their corresponding "given" instances) determine the immunity against or susceptibility of a voting system to the corresponding control. In addition, if a result shows that a voting system is susceptible to a specific type of control then it also shows the level of this susceptibility in terms of computational complexity, that is whether the voting system is *computationally* resistant to control or *computationally* vulnerable to control. To fully grasp the meaning behind these terms we need clear definitions of the terms immunity, susceptibility, resistance and vulnerability. All of these terms are defined similarly for both constructive and destructive control as follows.

Immunity: a voting system is immune to constructive control through a given method if it is never possible for the authority running the election to promote a non-winner candidate to becoming the unique winner by using the control method. Similarly, a voting system is immune to destructive control through a given method if it is never possible for the authority running the election to prevent a winning candidate from becoming the unique winner by using the control method.

Susceptibility: if a voting system is not immune to a model of control then it is said to be susceptible to that model of control.

Vulnerability: a voting system is (computationally) vulnerable to a (model of) control if it is susceptible to that (model of) control and the corresponding problem is computationally easy to solve (i.e., it is in P).

Certifiable Vulnerability: a voting system is certifiably vulnerable to a model of control if it is vulnerable to that model of control and an algorithm that the chairman needs to implement to control the outcome of the election exists and runs in polynomial time.

Resistance: a voting system is resistant to a control model if it is susceptible to control but the corresponding control problem is computationally hard (NP-complete [6, 36] or NP-hard [37, 29]).

The first two properties mostly result from knowledge in axiomatic social choice theory and political science. The last two mostly stem from results derived from complexity-theoretic treatments of the problem. The two however are related. Since general susceptibility results were shown by studies in the theory of social choice and economics, this inspired more specific and practical results based on some new tool and a new perspective;

which is complexity-theory. The section on "Proofs" touches on how the results were obtained, in terms of the technique used, and provides a summary of findings. Before getting there, it is now time to discuss an important work in electoral conrol.

4.3.5 Hybridization of Voting Systems and Control

As previous work showed, none of the studied election systems is globally resistant to all types of control. This motivated the question of whether or not there exists an election system with the property of being resistant to all twenty types of control[20]. In "Hybrid elections broaden complexity-theoretic resistance to control," Hemaspaanra et al., showed that there exists such a system [37]. This system is basically a hybrid of a number of voting systems such that it inherits the easiness of implementation (i.e., the winner(s) can be determined in polynomial time) if all constituents systems are easy to implement yet it also inherits the difficulty of control if at least one of the underlying systems is hard to control. The authors described the "hybridization" in their work as an analog to the "join" operation discussed in the previous section on complexity theory. The combined result of resistance to control and easiness of implementation was derived by the way inheritance works when the join operation is used. This work proved, for the first time, that a desirable election system (though artificial) indeed exists.

4.3.6 Proofs

The proof techniques vary and seem to be strongly connected to the type of results. Immunity results (mostly) require direct proof methods based on the axioms of social choice theory and political science, and knowledge of results in these former fields. Since susceptibility is merely the logical negation of immunity, the proofs concerning these results are somewhat related. On the other hand, proofs of vulnerability results vs. resistance employ techniques from computer science, in particular algorithm design and complexity theory. In specific, for vulnerability results the proof basically provides an algorithm (well-defined step by step specification) for answering the yes/no question of the given instance and then argues that the algorithm runs in time that is polynomial in the length of the input. For certifiable-vulnerability results, the proof provides an algorithm, i.e., actions to be taken by the chairman, to control the election and then argues that the algorithm runs in polynomial time (solves the problem whose decision version is proven easy in the vulnerability case). Finally, for resistance results the special work of complexity theory comes into play and corresponding language problems are shown to be NP-complete or NP-hard via reductions from known NP-hard problems (and polynomial verification for membership to NP in case of NP-completeness results). Many NP-hard results were obtained by reductions from different variants of the Exact-Cover NP-complete problem.

The following are selected theorems and their proofs. The results and proofs—except for Theorem 4.3.3 which is proven independently here—are due to previous work as cited. The proofs are rewritten here.

[20] Except of course control by effective sequence of pairwise comparisons, since this one only applies to the sequential pairwise voting rule.

Theorem 4.3.1. *Condorcet voting is immune to constructive control by adding candidates [6].*

Proof. If a given candidate is not a winner under the *uncontrolled* election, then by definition of the Condorcet rule that candidate is defeated by at least one other candidate. Adding new candidates will not change the fact that c is defeated by some candidate and hence will not make c the winner of the new election. □

Theorem 4.3.2. *Plurality voting is computationally vulnerable to constructive control by deleting voters [6].*

Proof. Since Plurality considers only first choice candidates, each vote influences the position of one candidate only. Hence, it is possible to delete voters in order to promote a candidate to be the winner. □

Theorem 4.3.3. *Plurality voting is certifiably vulnerable to destructive control by deleting voters.*

Proof. Since Plurality considers only first choice candidates, the algorithm to prevent a given candidate c from winning is simply as follows: consider the number of voters choosing c as a top candidate. Finding this is possible in polynomial time since there is a total of $\|V\|$ voters whose preferences are known to the chairman in advance. If this number is at most k, then there exists a set of k voters (where $k \leq \|V\|$, see defitnition 4.3.9) whose removal would guarentee that c is not the winner. Otherwise, there does not exist a set of k voters whose removal would guarentee that c is not the winner. This algorithm is polynomial in the number of voters (which is clearly a polynomial in the size of the input). □

Theorem 4.3.4. *Sequential pairwise majority voting is certifiably vulnerable to control by effective sequence of comparisons [6].*

Proof. The following is a polynomial-time algorithm that determines whether a sequence of pairwise competitions exists such that a given candidate c wins the sequential election under this sequence, and if it exists, the algorithm finds that sequence in polynomial time.
1. Construct the tournament G of all pairwise comparisons between candidates[21].
2. Use breadth-first-search to determine whether a spanning tree (that spans all candidates) rooted at c exists.
3. If the tree in step 2 is found, then c is the winner of election given the sequence in which candidates appear in non-increasing order of their distance from c in G. Otherwise, some other candidate (not in the *maximal* tree rooted at c wins the election).
Step 1 is $O(|V||C|^2)$, step 2 is $O(|C|)$ and the construction of the tree is $O(|C|^2)$. The algorithm is $O(|V||C|^2)$ which is clearly polynomial in the length of the input. □

[21] A tournament is defined in Chapter 2.

4.3.7 Results

Results obtained from work on the control problem show that most natural and widely used voting systems are not globally immune or resistant to control [6, 37, 29]. Nevertheless some are more susceptible than others to specific model of control. For example, Plurality is resistant to structure control of the candidate set but it is vulnerable to control of the voter set. The Condorcet method, to the contrary, is mostly vulnerable to constructive control of the candidate set but it is resistant to control of the voter set. The only voting system that is resistant to all twenty standard types of control[22], is an artificial one. Nevertheless, two natural voting systems, namely Copeland and Llull's were shown to be "broadly" resistant to control. Moreover, tie-eliminating rules play an important role in the complexity of control by parition of candidates. It is worth mentioning too that the results obtained for Llull's system apply also to irrational voter set, since his system was defined on possibly non-transitive preferences.

The following table summarizes the results for the different control models for a number of voting systems. *hybrid* stands for the hybrid election system described in [37].

[22]Not counting control by effective sequence of comparisons because of its restricted application.

Table 4.3: Summary of results on the complexity of electoral control.

System	Plurality		Condorcet		Approval		*hybrid*		Llull		Copeland	
Model	C	D	C	D	C	D	C	D	C	D	C	D
AC	R	R	I	V	I	V	R	R	V	V	V	V
DC	R	R	V	I	V	I	R	R	R	V	R	V
PC	TE: R TP: R	TE: R TP: R	V	I	TE: V TP: I	TE: I TP: I	R	R	TP: R TE: R	TP: V TE: V	TP: R TE: R	TP: V TE: V
RPC	TE: R TP: R	TE: R TP: R	V	I	TE: V TP: I	TE: I TP: I	R	R	TP: R TE: R	TP: V TE: V	TP: R TE: R	TP: V TE: V
AV	V	V	R	V	R	V	R	R	R	R	R	R
DV	V	V	R	V	R	V	R	R	R	R	R	R
PV	TE: V TP: R	TE: V TP: R	R	V	TE: R TP: R	TE: V TP: V	R	R	TP: R TP: R	TP: R TP: R	TP: R TP: R	TP: R TP: R

Key: C = Constructive, D = Destructive,
AC = Adding Candidates , AV = Adding Voters,
DC = Deleting Candidates , DV = Deleting Voters,
PC = Partitioning of Candidates, RPC = Run-off Paritioning of Candidates, PV = Partioning of Voters,
I = Immune, R = Resistant, V = Vulnerable,
TE = Ties-Eliminate and TP = Ties-Promote.

The following are additional results that are not mentioned in the table above:

Theorem 4.3.5. *Both Llull's voting system and Copeland voting system are resistant to constructive control by adding $\leq k$-candidates and both are vulnerable to destructive control by adding $\leq k$-candidates [29].*

Theorem 4.3.6. *For SNTV,* Multi-Winner-Control *(by adding voters) is in P [56]*

Theorem 4.3.7. Multi-Winner-Control*(by adding voters) is* NP*-complete for Bloc voting, Approval voting, and Cumulative voting [56].*

4.3.8 Comments and Bibliographic Notes

- The scope of the related research works discussed in this section is the control problem from a complexity-theoretic perspective. Although the scope is basically the same for all of these studies, the focus of each is clearly different. The original work of Bartholdi, Tovey and Trick focused on the complexity of *constructive* control for Plurality and Condorcet voting systems. Following their work was the work of Hemaspaandra, Hemaspaandra and Rothe on the complexity of *destructive* control for plurality, Condorcet and approval voting systems. The latter introduced tie-breaking and refined complexity results of control by partition of candidates based on the tie-breaking rule in use. A notable work of Hemaspaandra et al. established the existence of a voting system that is resistant to all types of control discussed previously [37]. A work of Faliszewski et al. introduced yet positive results regarding resistance to control by Copeland and Llull's voting systems [29]. In addition, this work studied a restricted form of control by adding a *bounded* number of candidates.

- In the context of multi-winner systems, Procaccia et al. redefined the control-by-adding-voters problem for multi-winner systems using utility functions. New results were obtained based on this definition for Bloc[23]voting, Approval, SNTV, and Cumulative voting [56].

[23]Chapter 1 provides definitions of these voting rules.

4.4 The Complexity of Manipulating Elections

4.4.1 Background

In the voting context, manipulation stands mainly for a voter's misreporting his/her vote such that it does not reflect true preferences, so that to change the outcome of the election as desired. This is also referred to as strategic voting in which agents or voters are called "self-interested" agents. When interested in a specific outcome of the election, the "would-be" manipulator asks whether there exists an "effective preference" that would tailor the outcome of the election to be the favorable one. Is it clear then that some researchers refer to manipulation in election as the problem of finding an "effective preference."

For example, consider the sequential pairwise majority rule. Assume that the election consists of three candidates x, y and z, and three voters v_1, v_2 and v_3. The agenda is as follows: x and y compete in the first round, and the winner competes with z in the second round. Assume that the voters have the following (true) preferences respectively: $x > y > z$, $y > z > x$, and $z > x > y$. Under the sequential majority rule, the winner of the first round is x so x competes with z next, and according to the preferences above, z wins. Note that z is v_1's least preferred candidate. If instead of reporting her (true) preference $x > y > z$, v_1 reports the following preference order: $y > x > z$, then the winner of the first round is y and then y competes with z in the second round and wins. So by reporting a preference order other than his true preference order, v_1 achieves an outcome (y) that he prefers to the outcome under true preferences (z). We say in this case that the sequential pairwise majority is manipulable, or it is not strategy-proof, since there is a voter who can manipulate the rule by misrepresenting preferences to achieve a desirable outcome.

Since voting systems are supposed to output a socially desirable choice, misrepresenting a vote can lead to an outcome that does not reflect the true preferences of the society members. Therefore, manipulation is considered an undesirable phenomenon in voting. While this is a concern in human contexts, it has a similar if not a higher importance in software agents settings. This is because software agents are not affected by irrationality, emotions or limited computational speed and capacity. Therefore, computing effective preferences can be done easily in societies of software agents.

It was known for a long time that *many* voting procedures are subject to manipulation by strategic voters. The first conjecture on the manipulability of voting procedures *in general* was Dummett and Farquharson's conjecture, in which they state: "It seems unlikely that there is any voting procedure in which it can ever be advantegous for any voter to vote 'strategically', i.e., non-sincerely" [23], page 34.

Vickery [61] made another conjecture based on Arrow's theorem. Basically, he conjectured that satisfying the conditions of IIA and Pareto efficiency, implies manipulability or dictatorship. The conjecture was proven independently by Gibbard [32] and Satterthwaite [59]. The result is referred to as Gibbard-Satterthwaite theorem. Gibbard-Satterthwaite's theorem and Duggan-Schwartz's theorem apply to voting rules with at least three candidates, but the first one applies to single-winner rules while the second applies to rules that output a set of winners. The two theorems prove that there does not exist a non-dictatorial non-manipulable voting system without compromising other conditions of fairness and uniformity.

Basically, these theorems establish the theoretical impossibility of non-manipulability of voting systems. However, like Arrow's theorem, these therorems show that there are cases where voting systems (satisfying the conditions of the theorems) are manipulable. It does not say that every election is mainpulable. Moreover, these results do not tell much about the practical difficulty of manipulating a specific voting rule, i.e., if it is true that voting systems are basically manipulable, then is it feasible in practise to find effective preference that will change the outcome of the election if reported? This question was first raised by the pioneers of computational voting theory: Bartholdi, Tovey and Trick in their paper: "The Computational Difficulty of Manipulating an Election" [4] and then again by Bartholdi and Orlin in "Single Transferable Vote Resists Strategic Voting" [3]. Since then, the research in the computational complexity of manipulation has been actively and rapidly extending. The manipulation problem is the most extensively studied in the complexity theorey literature of voting systems. The following sections shed some light on some of the main results in this area.

4.4.2 Different Manipulation Problems

The followings are different formulations of the manipulation problem:

Definition 4.4.1 (Existence of a Winning Preference)**.** *Given a set of candidates C, a set of votes*[24] *V that are transitive and complete orders over C, and a distinguished candidate $c \in C$.*
Question*: Does there exist a preference order P that will ensure that c is the winner?*

This is a formulation of a single-voter manipulation problem (which appeared in [4]).

Definition 4.4.2 (Effective Preference)**.** *Given a set of candidates C, a set of votes V that are sincere, transitive and complete orders over C, and a distinguished candidate $c \in C$.*
Question*: Does there exist a preference order on C that when tallied with the preferences of V will ensure that c is the winner?*

This is also a formulation of a single-voter manipulation problem (which appeared in [3]).

Definition 4.4.3 (Preferred Outcome)**.** *Given a set of candidates C, a set of votes V that are sincere transitive and complete orders over C, and the sincere preferences of the manipulator.*
Question*: Can the manipulator achieve a preferred outcome in the election by voting other than his/her true preferences?*

This is also a formulation of a single-voter manipulation problem but it focuses on finding a strategic vote that will yield results better than the sincere vote (appeared in [3]).

[24] or voters having votes that are reported as transitive and complete orders over C.

Definition 4.4.4 ($\mathcal{E}$-manipulation)**.** *Given a set of candidates C, a set of non-manipulative voters V, a set of manipulative voters S with $V \cap S = \emptyset$ and a distinguished candidate $c \in C$.*
Question: *Is there a way to set the preferences lists of the voters in S such that c is the winner of the election $(C, V \cup S)$ under the rule $\mathcal{E}$?*

This is a formulation of a coalition-of-voter manipulation problem.

The following are definitions of constructive manipulation by a coalition of weighted voters for deterministic and randomized protocols respectively [12].

Definition 4.4.5 ($\mathcal{E}$-Constructive-Coalitional-Weighted-Manipulation–CCWD)**.** *Given a set of candidates C, a set of non-manipulative weighted voters V, the weights for a set of manipulative voters S with $V \cap S = \emptyset$ and a distinguished candidate $c \in C$.*
Question: *Is there a way to cast the votes in S such c is the winner of the election $\mathcal{E}(C, V \cup S)$?*

Definition 4.4.6 ($\mathcal{E}$-Constructive-Coalitional-Weighted-Manipulation–CCWR)**.** *Given a set of candidates C, a set of non-manipulative weighted voters V, the weights for a set of manipulative voters S with $V \cap S = \emptyset$, a distinguished candidate $c \in C$, a distribution over instantiations of the voting protocol, and a number r, where $0 \leq r \leq 1$.*
Question: *Is there a way to cast the votes in S such c is the winner of the election $\mathcal{E}(C, V \cup S)$ with probability greater than r?*

The following are definitions of destructive manipulation by a coalition of weighted voters for deterministic and randomized protocols respectively [12].

Definition 4.4.7 ($\mathcal{E}$-Destructive-Coalitional-Weighted-Manipulation–DCWD)**.** *Given a set of candidates C, a set of non-manipulative weighted voters V, the weights for a set of manipulative voters S with $V \cap S = \emptyset$ and a candidate $c \in C$.*
Question: *Is there a way to cast the votes in S such c is not the winner of the election $\mathcal{E}(C, V \cup S)$?*

Definition 4.4.8 ($\mathcal{E}$-Destructive-Coalitional-Weighted-Manipulation–DCWR)**.** *Given a set of candidates C, a set of non-manipulative weighted voters V , the weights for a set of manipulative voters S with $V \cap S = \emptyset$, a candidate $c \in C$, a distribution over instantiations of the voting protocol, and a number r, where $0 \leq r \leq 1$.*
Question: *Is there a way to cast the votes in S such c is the winner of the election $\mathcal{E}(C, V \cup S)$ with probability less than r?*

4.4.3 Types of Manipulation

There are different factors to be considered when analyzing the complexity of manipulating elections. These include the nature of the manipulation, i.e., whether constructive or destructive, whether manipulation is done by a single voter or a group of voters and whether the votes are weighted or not. The following few paragraphs briefly explain these factors.

Manipulation can be effected by a single voter or a number of voters (a coalition of voters).

Like electoral control, manipulation can be constructive or destructive. In constructive manipulation, a voter or a coalition of voters misreports preferences so that a distinguished candidate c becomes the winner of the election. In destructive manipulation, a voter or a coalition of voters misreports preferences so that a distinguished candidate c does not win the election.

In weighted manipulation, a voter has a weight and so the influence of his/her vote on the election's outcome is related to the weight the voter has. In elections where votes are unweighted, two voters with the same ordering over the candidates contribute equally to the rank of the alternatives in the final result of the election. A k-weighted vote can be viewed as k unweighted votes and hence many results pertaining to weighted voting/manipulation are used to infer results in the unweighted version of the problem.

The type of manipulation to be studied is determined by some or all of the previous factors. In the following sections, we will encounter general results and results that are specific to each type of manipulation.

4.4.4 Questions and Directions

The following are the main questions that led different directions in the area of manipulation complexity:

- What is the computational complexity of manipulating multi-winner voting schemes?
- What is the computational complexity of manipulation for some common voting rules?
- How is the complexity of manipulating elections related to the number of candidates?
- Is it possible to design voting rules that are hard to manipulate? If so, how?
- What is the manipulation power of voter coalitions?
- What is the source of the complexity of manipulation (with respect to some broad family of voting protocols)? In other words, what are the characteristics of a voting system that determine its manipulation complexity?
- Are there manipulation related analyses other than worst-case?
- What is the computational complexity of manipulating voting rules under incomplete knowledge of preferences?

Each of the following subsections is dedicated to the discussion of one of these directions and is concluded with answers for related questions.

4.4.5 Complexity of Manipulating Multi-Winner Schemes

Recall the definitions of the following k-winner voting schemes: pproval, Bloc Voting, Cumulative Voting and SNTV. Manipulation for these schemes is defined in [56] as follows:

Definition 4.4.9. *Given a set C of candidates, a set V of voters that have already cast their votes, the number of winners $k \in \mathbb{N}$, a utility function $u : C \to \mathbb{Z}$ and an integer $t \in \mathbb{N}$.* ***Question:*** *Can the manipulator cast its vote such that in the resulting election[25]: $\Sigma_{c \in W}\, u(c) \geq t$, where W is the set of winners and $\|W\| = k$.*

The following are results pertaining to manipulating the abovementioned multi-winner schemes. These results are due to Procaccia et al. [56].

Theorem 4.4.1. *Manipulation in Approval voting, Bloc voting, and SNTV is in* P.

The proof was omitted in the paper, but basically the algorithm is simple. The manipulator awards points (according to the rule) to candidates whom he wants to win exclusively. If this suffices to make his favorite candidates win then the vote was effective, otherwise we conclude that there is no way the manipulator can maximize the utility of favorite candidates as required by awarding all points to these candidates only.

Theorem 4.4.2. *Manipulation in Cumulative voting is* NP*-complete.*

Membership in NP is easy, hardness was obtained via a reduction from Knapsack.

4.4.6 Complexity of Manipulating Common Voting Rules

For an unbounded number of candidates and voters, and under the assumption of complete information about other voters' preferences, the following are results regarding constructive manipulation as defined in *Existence of a Winning Preference*:

Theorem 4.4.3. *Plurality, Borda Count, Maximin, and Copeland's method are all manipulable in polynomial time.*

Theorem 4.4.4. *Any monotone increasing function of voting methods that are all manipulable in polynomial time is also manipulable in polynomial time.*

Second-order Copeland stands for a tie breaking rule used with the Copeland method that has a drawback of producing many ties. In second-order Copeland, if a tie occurs then the winner is "the candidate whose defeated competitors have the largest sum of Copeland scores" [4], page 231.

Theorem 4.4.5. *Manipulation of second-order Copeland is* NP*-complete.*

Theorem 4.4.6. *Manipulation of first-order Copeland with second-order Copeland tie breaks is* NP*-complete.*

Theorem 4.4.7. *Manipulation of Single Transferable Vote[26] is* NP*-complete.*

[25] The authors in [56] remark that tie breaking is assumed to go against the manipulator. So if a number of candidates do equally well, then the candidates with the lower utility win. This is a standard assumption in handling tie breaking for manipulation problems.

[26] There are different variations of STV, here STV is defined as in Chapter 1. The method is also called *successive elimination* or *alternative voting.*

The first four results are due to Bartholdi, Tovey and Trick [4], and the last one is due to Bartholdi and Orlin [3]. The results listed above assume complete knowledge about all voters' preferences. One can argue that if it is hard to manipulate a voting scheme given complete knowledge of others' preferences, then manipulation of the scheme will also be hard given incomplete information. However, polynomial time results will be restricted to this assumption, we will discuss later the manipulation problem under incomplete knowledge of preferences.

The above results were obtained for unbounded number of candidates and voters. In many settings, however, there are restrictions on the number of candidates and sometimes even on the number of voters or typically on the ratio of the number of candidates to voters. In presidential elections for example, there are many more voters than candidates while in some search engine settings the number of candidates (e.g., webpages) far exceeds the number of voters (e.g., other search engines). The following section spells out results of manipulation complexity for bounded number of candidates.

4.4.7 Manipulation of Elections and the Number of Candidates

The initial work studying the manipulation problem did not explicitly consider the number of candidates when addressing the complexity of manipulation. It was assumed that the number of candidates is unbounded. Conitzer and Sandholm analyzed first the complexity of manipulating elections with few candidates [13], a following paper by the same authors appeared a year later and answered the question of how many candidates (minimum) are needed to make manipulation hard [12]. These results together with new ones were presented in a following work by Conitzer et al. [19].

Some of the assumptions made were similar to previous works assumptions and some where different. The authors considered coalitions of weighted voters and showed that for simpler voting settings manipulation is easy for a small number of candidates. Also, it may be more realistic and important to study the effectiveness and efficiency of coalition manipulation. Results obtained for this model can be used to infer results for different settings. For example, results obtained for a coalition of manipulators with complete information can be used to get results for single manipulator with incomplete information setting. Similarly, results pertaining to weighted voters in the complete information setting can be used to find results for the unweighted limited information setting.

The following tables summarize the results of the first and second papers ([13] and [12]) grouped by constructive and destructive manipulation types respectively:

Table 4.4: Summary of results on the complexity of constructive manipulation of elections with few candidates by a coalition of weighted voters.

Number of candidates	2	3	4,5,6	⩾7
Plurality	P	P	P	P
Regular Binary Cup (BC)	P	P	P	P
Randomized BC	P	P	P	NP-complete
Copeland	P	P	NP-complete	NP-complete
Maximin	P	P	NP-complete	NP-complete
Borda	P	NP-complete	NP-complete	NP-complete
lurality with run-off	P	NP-complete	NP-complete	NP-complete
STV	P	NP-complete	NP-complete	NP-complete
Veto	P	NP-complete	NP-complete	NP-complete

Table 4.5: Summary of results on the complexity of destructive manipulation of elections with few candidates by a coalition of weighted voters.

Number of candidates	2	⩾3
Borda	P	P
Copeland	P	P
Maximin	P	P
Plurality	P	P
Veto	P	P
Regular Binary Cup (BC)	P	P
Randomized BC	P	?
Plurality with run-off	P	NP-complete
STV	P	NP-complete

We observe that Plurality is the easiest to manipulate while STV, Borda, Veto and Plurality with Run-off are the hardest to manipulate when the number of candidates is a small constant. This also draws attention to the complexity of STV, Veto and Plurality with Run-off. Notice that in all of these rule, determining winners is a multi-phase process. With this in mind we are ready to introduce the next subsection.

4.4.8 Designing Voting Protocols that are Hard to Manipulate

Some research work in the complexity of manipulation shifted from the study of the complexity of manipulating existing voting systems to the topic of designing new voting systems that are hard to manipulate. There are two main works to be discussed here: one proposes modifying existing systems and the other proposes hybridization of voting systems. The goal in the two approaches is the same, to raise the complexity of manipulation.

I. Modifying Existing Systems

The first work in this area suggested modifying existing efficiently manipulable voting schemes in such a way that the resulting system is computationally difficult to manipulate. In their paper titled "Universal voting protocol tweaks to make manipulation hard," Conitzer and Sandholm show how a simple modification (tweak), can dramatically raise the complexity of manipulating a voting system from being in P to being NP-complete or even higher [15]. The tweak they suggested is simply the addition of an elimination preround to the election. Before we explain this preround, we highlight reasons the authors mentioned for choosing to modify existing voting protocols as a step towards designing new hard-to-manipulate protocols:

- Complexity results obtained from this modification apply to a large family of voting protocols and are not restricted to a specific voting protocol (in comparison, designing a new protocol will mostly imply results pertaining to that particular protocol).
- The modifications do not significantly change the original work, and this has two advantages: first, it preserves the desirable theoretical properties of the original system. Second,
- it is much easier in practice to work with a slightly modified known voting system than to work with a totally new one.

I.1 The Pre-round Modification

Definition 4.4.10. *Given a voting rule $\mathcal{E}$, the new (modified) protocol proceeds with an initial preround as follows:*

1. *The candidates are paired. If there is an odd number of candidates, a candidate gets a bye.*
2. *Each pair of candidates compete in a pairwise election, the candidate who loses is eliminated. (the candidate who got a bye is never eliminated).*
3. *$\mathcal{E}$ is executed on the remaining candidates (including the one who got a bye) to produce a winner*[27]*.*

I.2 Manipulation of Modified Protocols

The complexity of manipulating the modified system depends on whether the pairing (also called scheduling) is done and known to all voters before votes are drawn, whether the pairing is drawn completely randomly after the voters vote, or whether the two processes of pairing and voting are interleaved. According to the previous description of when the pairing is done, the preround is called deterministic, randomized or interleaved respectively.

This work considered constructive manipulation by a single voter with all other votes

[27] Eliminated candidates are dropped from the votes. For example if a voter voted $a > b > c > d$ and b and c are eliminated, then the voter's "implied" vote is $a > d$.

known to the manipulator. In each of the aforementioned protocol types, the the question asked is:

- Can the manipulator cast the vote to make c win the election under $\mathcal{E}$?
- Can the manipulator cast the vote to make the probability of c winning the election under $\mathcal{E}$ at least equal to k where k is some given value $\in [0, 1]$?
- Given the initial random choices (if any) by the protocol, is there a contingency plan (based on the random decisions the protocol takes between collecting parts of the votes) for the manipulator to answer the queries to make the probability of c winning under $\mathcal{E}$ at least equal to k where k is some given value $\in [0, 1]$?

I.3 Results

Theorem 4.4.8. *The following protocols when preceded with a deterministic preround (the first type of pairing of alternatives) are* NP*-complete to manipulate: Plurality, Borda, Maximin and STV.*

Theorem 4.4.9. *The following protocols when preceded with a randomized preround (the second type of pairing of alternatives) are* #P*-complete to manipulate: Plurality, Borda, Maximin and STV.*

Theorem 4.4.10. *The following protocols when preceded with an interleaved preround (the first type of pairing of alternatives) are* PSPACE*-complete to manipulate: Plurality, Borda, Maximin and STV.*

Remark: This is the first work in computational voting theory that have results in a complexity class that is higher than NP (assuming PSPACE $\neq$ NP).

I.4 An Alternative Modification

Inspired by the pre-round modification and by the open question of whether this approach can make manipulation as hard as inverting one-way functions, Elkind and Lipmaa proposed a pre-round where the schedule is computed from the votes using a one-way function [26]. They argued that for many reasons this one-way function approach has appeal over the pre-round approach explained above. For security reasons, the pre-round schedule should not be determined by the authority conducting the election but should be extracted fairly based on all votes. This approach extracts randomness from the votes themselves. For conceptual reasons, the problem of manipulation in voting resembles the cryptographic challenge of designing protocols that are resistant to malicious attacks with reliable probabilities. One-way functions have been proposed for that purpose in the cryptography domain.

Although this approach has produced interesting hardness results for manipulation by small coalitions of voters, its hardness results apply only to settings with a very large number of candidates and it is restricted to given fractions of the voter set as a coalition size (another work discussed the asymptotic average size of effective coalition of manipulating voters [54]).

II. Hybridization of Voting Systems

Based on the observation that the pre-round in the previous work is basically the first step of the binary cup (BC) rule, Elkind and Lipmaa constructed another family of voting systems that are hard to manipulate [25]. They used *hybridization* of voting systems to construct new protocols that are hard to manipulate.

They defined hybridization of two voting systems X and Y to be the voting system that starts by executing X—mostly to eliminate a number of candidates—then to execute Y on the remaining candidates, in a sequential manner[28]. X is usually defined as the k-steps of some known voting system and Y is also one of the common voting systems. A step is a single stage of the protocol and it depends on the particular protocol. In STV, for example, a step consists of eliminating a candidate with the lowest plurality score, while a step in BC consists of pairing up the candidates and eliminating the ones who lose the pairwise election against their opponents. The hybrid of k-steps of Y, and X is denoted $Hyb(X_k, Y)$ and is formally defined as follows:

Definition 4.4.11. *A hybrid protocol $Hyb(X_k, Y)$ consists of two phases. In the first phase, the protocol executes k steps of X on all candidates; suppose that S is the set of candidates not eliminated in the first phase. In the second phase, the protocol applies Y to the implied votes on S, i.e., votes restricted to the remaining set of candidates S.*

Inspired by the results of the previous work of Conitzer and Sandholm [15], this work focused on creating a more general framework for constructing voting systems that are hard to manipulate. Unlike the case of the pre-round modification with regard to when votes are drawn, here the execution of the voting rules takes place entirely after casting the votes. After the elimination of some candidates however, the scores are recomputed but this is done on basis of the initial votes. In addition to this difference, the authors of "Hybrid Voting Protocols and Hardness of Manipulation" mentioned that although the pre-round approach preserved some properties of the modified system (such as Condorcet-consistency), eliminating half or a significant percentage of the candidates using a criteria that may be fundamentally different in spirit from the original protocol, may result in a considerably different mechanism and outcome (which can be both not desirable and can—expectedly—cause a considerable change in the complexity of the manipulation) [25]. Their approach, however, can be used to construct hybrids of a voting system with itself or with some other chosen voting system without changing the function of the constituent systems.

This work also showed that not all hybrids are NP-hard to manipulate, for instance $Hyb(Plurality_k, Y)$ where $Y \in \{Plurality, Borda, Maximin, BC\}$ can be manipulated in polynomial time, but on the other hand this work successfully used the hybridization framework to show that many hybrids of common voting systems are hard to manipulate. The following list summarizes these results:

[28]This is not the same definition of hybridization that was used to construct a voting system that is resistant to electoral control in the previous section.

Hybridization Results

Theorem 4.4.11. *The hybrids of the form* $Hyb(STV_k, Y)$ *where* $Y \in \{Plurality, Borda, Maximin, BC, STV\}$ *are* NP-*hard to manipulate for infinitely many values of* k.

Theorem 4.4.12. *The hybrids of the form* $Hyb(X_k, STV)$ *where* $X \in \{Plurality, Borda, Maximin, BC\}$ *are* NP-*hard to manipulate for infinitely many values of* k.

Theorem 4.4.13. *The protocols* $Hyb(Borda_k, Plurality)$ *and* $Hyb(Maximin_k, Plurality)$ *are* NP-*hard to manipulate for infinitely many values of* k.

Theorem 4.4.14. *The (hybrid) protocol* $Hyb(Borda_k, Borda)$ *is* NP-*hard to manipulate for infinitely many values of* k.

Theorem 4.4.15. *The (hybrid) protocol* $Hyb(Maximin_k, Borda)$ *is* NP-*hard to manipulate for infinitely many values of* k.

In addition to these computational attributes of hybrid protocols, the following result summarizes other attributes of hybrid protocols[29]:

Proposition 4.4.1. *For any voting systems* X *and* Y *and any* k,

- *if both* X *and* Y *are Condorcet-consistent then so is* $Hyb(X_k, Y)$.
- *if* X *is strongly Pareto-optimal and* Y *is Pareto-optimal then* $Hyb(X_k, Y)$ *is Pareto-optimal.*
- *if* X *is strongly monotone and* Y *is monotone then* $Hyb(X_k, Y)$ *is monotone.*

In addition to these results, a result similar was obtained for utility-based voting which suggests that the technique of hybridization can be used for even a wider class of preference aggregation protocols. We draw attention to the contribution of this work in understanding what makes some voting protocols inherently manipulation-resistant (in light of results given in previous subsections, think of hybridization and compare STV to Borda for example). An important work in the manipulation literature addressed the topic of what characteristic determines the degree of resistance a voting system have against manipulation. This is the topic of the next subsection.

4.4.9 What Makes Voting Rules Hard to Manipulate

As stated generally, the question "What Makes Voting Rules Hard to Manipulate?" it is still an open question. However, for two general families of voting protocols, the answer is known. These two families are the family of *scoring protocols* and the family of *scoring elimination protocols*. Recall the definition of positional scoring protocols given in Chapter 1. The family of scoring protocols is defined by a vector $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ of integers

[29] Refer to Chapter 1 for definitions of the properties.

as a vote. The value of each α_i for $1 \leq i \leq m$ where m is the number of candidates, in the vector depends on the particular voting protocol. For example, for plurality, all α_i's are equal to 0 except for α_1 which is equal to 1. The points in all votes are then added for every candidate and the candidate with the maximum number of points wins.

We can think of positional scoring protocols as a generalization of plurality voting. Similarly, we can think of scoring elimination protocols as a generalization of STV. STV is a multi-round elimination protocol that uses plurality as the underlying scheme in each round. In general, any other positional scoring protocol can be used in the individual rounds, such as the Borda-count, or veto. For example, when the underlying protocol is veto, then the successive elimination rule using veto is the Coombs rule.

For both these families of voting rules, we have a dichotomy theorem for the hardness of manipulation.

For all positional scoring protocols, the basic manipulation problem by a single-voter is in P. Therefore, the first dichotomy theorem regards manipulation by a coalition of voters. The case is different for scoring elimination protocols since STV is known for example to be NP-complete to manipulate by a single voter in the general case.

The first dichotomy theorem is due to Hemaspaandra and Hemaspaandra [33, 34]. The theorem basically says that every positional scoring protocol that assigns two or more points to candidates other than the favorite candidate is NP-complete to manipulate. Every other scoring protocol is manipulable in polynomial time. This means that plurality protocol and plurality-like protocols (such as veto for example) are computationally easy to manipulate, and all other systems are computationally difficult to manipulate.

Theorem 4.4.16 (Dichotomy Theorem for Positional Scoring Protocols)**.** *For any scoring protocol $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ and any m, manipulation and weighted-manipulation by a coalition of voters are in* P *if $\alpha_2 = \alpha_3 = \ldots = \alpha_m$, and are* NP*-complete in all other cases.*

The theorem applies for a general number of candidates, for weighted and unweighted votes, and for the unique and non-unique winner cases.

The second dichotomy theorem is due to Coleman and Teague [9]. It applies to scoring elimination protocols with fixed number of candidates.

Theorem 4.4.17 (Dichotomy Theorem for Scoring Elimination Protocols)**.** *When the number of candidates is fixed, for any scoring elimination protocol $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ eighted-manipulation by a coalition of voters is in* P *if $\alpha_1 = \alpha_2 = \ldots = \alpha_{m-1}$, for all m, and is* NP*-hard in all other cases.*

4.4.10 Worst-case Hardness Results: a Second Look

One of the limitations of all hardness results discussed above is that they are all worst-case analysis. These results simply say that there are instances in which manipulation is going to be computationally difficult. They do not imply that every instance of manipulating the voting scheme for which hardness results were obtained is difficult to compute. The question is what if in practice most instances are not difficult to manipulate? Then hardness results have little practical weight. Another question is what is the average case complexity of manipulating voting rules? Progress in this direction was initiated by Procaccia and Rosenschein in "Junta distributions and the average-case complexity of manipulating elections" [55]. The average case analysis they introduce is not the typical (famously difficult) average case analysis. They use the concept of Junta distribution which is a distribution over NP-hard problem instances, to study this type of average case complexity of manipulating voting rules. A problem is shown to be computationally easy on average (i.e., in P) if a polynomial time algorithm can frequently find instances of the problem in the Junta distribution. They conjectured that such distribution exists for probabilistic indextermweighted manipulationweighted manipulation of voting protocols. They designed a heuristic polynomial time greedy algorithm to show that when the number of candidates is fixed, some scoring protocols such as the Borda count is easily manipulable on average by a coalition of voters (not only in the worst-case).

Conitzer and Sandholm raised a similar concern regarding worst-case hardness results. He argued that voting rules that are "usually" hard to manipulate do not exists unless other important desiderata are compromised [18]. The work presents this idea as an impossibility result that holds even for a stronger type of manipulation where the manipulator(s) needs to find the set of *all* candidates that they can promote to become winners. The work is concluded with suggestions to overcome this impossibility, such as extending the definition of voting rules, and allowing low-ranked candidates to win sometimes. One can argue that these modifications are not always feasible in practice which in turn strengthens the argument of nonexistence of usually hard to manipulate voting rules.

4.4.11 Manipulation Under Incomplete Knowledge of Preferences

Recall the previous discussions on vote elicitation and on finding winners under incomplete preferences. The same result apply to manipulation under *incomplete knowledge of preferences.* Since if an agent wants a specific alternative to win and plans to (strategically) vote in favor of this alternative then the agent must at least determine whether this alternative is a *possible* winner. Hence, the the problem of determining possible and necessary winners and the manipulation problem are closely related. This relation also stems from the fact the elicitation itself is closely related to the computation of possible and necessary winners, and this elicitation process can create opportunities for both the elicitor and agents to strategically influence the outcome based on knowledge about others agents preferences
General manipulability results concerning social choice functions on partial preferences have been obtained in [57], [51], and [52]. Recall the following terms from Chapter 3:

Strong dictator *A strong dictator is a voter such that, no matter how the others vote, this voter's ordering/choice is the outcome.*

Dictator *A dictator is a voter such that, no matter how the others vote, no outcome is selected outside the choice set of this voter.*

Weak dictator *A weak dictator is a voter such that, no matter how the others vote, some choices of this voter will always be included in the outcome.*

Given these definitions, the following are results concerning manipulability of social choice functions (and hence voting rules), under partial preferences:

Theorem 4.4.18. *If we have at least two agents and at least three alternatives, a social choice function on partial order without ties that is unanimous and monotonic has at least one weak dictator [51].*

Theorem 4.4.19. *If a social choice function is strategy proof and onto then it is unanimous and monotonic [52].*

Theorem 4.4.20. *If a social choice function is onto then it is either not strategy-proof or it has at least one weak dictator [52].*

Proof. From theorems 4.4.18 and 4.4.19 above. □

This result extends the Gibbard-Saterthwaite theorem to hold for social *choice* functions on *partial* preferences.

Results on Manipulation Under Incomplete Knowledge of Preferences

We conclude this section with a list of results related to manipulation under incomplete preferences:

Theorem 4.4.21. *In general, determining whether* $x \in PW_{\mathcal{E}}(R^n)$ *is* NP*-hard.*

Theorem 4.4.22. *In general, determining whether* $x \in NW_{\mathcal{E}}(R^n)$ *is* coNP*-hard.*

Theorem 4.4.23. *For a polynomially-computable voting rule* $\mathcal{E}$*, determining whether* $x \in PW_{\mathcal{E}}(R^n)$ *is in* NP*.*

Theorem 4.4.24. *For a polynomially-computable voting rule* $\mathcal{E}$*, determining whether* $x \in NW_{\mathcal{E}}(R^n)$ *is in* coNP*.*

Theorem 4.4.25. *For positional scoring protocols, possible and necessary winners can be computed in polynomial time [42]indexauthorsKonczak.*

Theorem 4.4.26. *For STV, computing possible winners is* NP*-complete and computing necessary winners is* coNP*-complete [53].*

Theorem 4.4.27. *It is* NP*-hard to return a superset of the possible winners* PW^* *under STV such that we guarantee* $\|PW^*\| < k\|PW\|$ *for some given positive integer* k *[53].*

Theorem 4.4.28. *Possible and necessary Condorcet winners can be computed in polynomial time [42]indexauthorsKonczak.*

Theorem 4.4.29. *For a given voting procedure, if necessary and possible winners are computed in polynomial time then deciding whether there is a (constructive/destructive) manipulation*[30] *is also possible in polynomial time [42]indexauthorsKonczak.*

[30] The complexity of manipulation will be discussed in some detail in the next chapter.

4.4.12 Comments and Bibliographic Notes

- In the previous chapter, we discussed single-peaked preferences and mentioned that under this restriction on the preference domain, voters cannot promote a non-winner alternative to become the winner under the new set of preferences. So manipulation in this sense is not possible in single-peaked domains. However, it is possible that the voters will change their votes such that no alternative wins the election. Using this section's terminology, *destructive* manipulation is possible even with single-peaked preferences.

- A related work that has not been listed under one of main directions discussed in this section is on manipulability when considering the number of manipulable profiles [48]. The minimal number of such profiles was established for voting rules with specific properties.

- We have discussed the computational complexity of the manipulation problem in this section. In the computing literature, manipulation of voting rules is usually used as a synonym for untruthful reporting of preferences, and typically has negative connotations. The terms strategic voting, manipulation, misrepresentation of preferences are used interchangeably. The argument here is against this inaccurate use of terms. Take for example the so called "manipulation" in approval voting, proposed "manipulation" algorithms suggest approving only candidates that the voter wishes to win, but then how can this act be described as "misrepresentation" of preferences? It is indeed a *representation* of the voter preferences. The counterargument might be: what if the voter *approves* other candidates but reports only a subset of those candidates in the vote in order to achieve a specific outcome? But then the reported vote represents the preferences of the voter at the voting time given feasibility concerns. The argument can be traced back to the philosophical meaning of "preference" and whether a vote as defined in voting rules, should be considered the same as preferences or as a way to represent those preferences whether explicitly or indirectly.
  The argument "if every voter misrepresent her preferences, then the result might be a socially undesirable outcome" (or something similar in essence) is usually presented when introducing the manipulation problem and discussing the motivation behind studying it. The argument here is that no such thing as "socially desirable outcome" exists! Given Arrow's theorem and other results on preference aggregation, we conclude that society "as a whole" does not have preferences, does not think (in the first place to think rationally second) as individuals perceive thinking and hence no outcome can be accurately described as "socially desirable outcome." All voting rules that we know are simply "attempts" to achieve an outcome that reflects the preferences of individuals in the society as much as possible. Yet no rule, is proven to achieve satisfactory results, even in the limited sense of satisfying a set of basic criteria. Moreover, one can argue (in a similar way) that if every voter votes strategically then the outcome is maximized welfare for all voters.

  We point out that literature on computational social choice, sheds more light on manipulation and whether it is a negative phenomenon or not. One particular work of interest here is by Hees and Dowding "In Praise of Manipulation." In this work, the authors (coming from philosophy and political science backgrounds) differentiate between different types of manipulation, which they call "sincere" and "transparent" manipulation and argue about the nature and the impact of each on voting outcomes. We suggest that computational treatments of the manipulation problem take into account the underlying meaning and implication of different variants of the manipulation problem and dedicate efforts into investigating the complexity of most relevant (undesirable) types.

4.5 The Complexity of Bribery in Elections

4.5.1 Introduction

The previous two sections shed light on the control and manipulation problems. This section discusses the problem of bribery in elections, mainly its computational complexity.

Bribery is similar to control in the sense that it needs to determine which voters to bribe in order to achieve a specific outcome. In the control problem, this party is the authority conducting the election whereas in bribery this party is any party (possibly a third party outside the election (i.e., neither candidates, nor voters nor the chairman) interested in changing preferences of voters.

Seen from a different perspective, bribery is also similar to manipulation in the sense that it involves changing voters' preferences. The difference between the two problems is the following: in manipulation, the set of manipulators is part of the input while in bribery it is part of the question. We will see in some of the following subsections that bribery is linked to manipulation and that results pertinent to the manipulation problem can be used to obtain new results for the bribery problem. This is a reason why this section on bribery comes after the sections on control and manipulation.

It is important to note that the computational complexity of a bribery problem is sensitive to the setting, and there are many different flavors of this problem. Therefore we start by defining different bribery problems next.

4.5.2 Different Bribery Problems

There are different settings for the bribery problem, the following list explains these settings:

- **Basic**: in the basic setting, voters are unweighted (equivalently, they all have the same weight and hence weights are ignored), and they all have the same price which is equal to the unit price. In this setting we are interested in the *number of voters* to bribe to achieve a specific outcome. This number is equal to the bribery budget in this basic setting.

- **Weighted**: in this setting, voters may have different weights, so voters with more weight are likely to influence the outcome more than voters with less weight.

- **Priced**: in this setting, voters have different price tags, so some voters are more expensive than others to bribe. In this case, the budget is the bribery *amount* to be spent to achieve a specific outcome. It is also possible that each voter has a different price tag depending on the degree to which her vote is to be influenced, but since it is unclear how to encode a voter's price scheme, this more complex setting has not been addressed in related work.

- **Weighted and Priced**: in this setting, voters have both weights and prices. It is natural to find voters with more weight also to have higher price tags. Nevertheless, the two entities (weight and price) are treated independently.

- **Input representation**: this refers to identical voters (same weight, price, and preferences) being represented by a single input next to a count in binary indicating the number of such voters. This is referred to as a "succinct" input. The non-succinct input refers to all voters being listed in the input. For example, if 100 voters are identical then there are 100 identical entries in the election input. Note that if $\mathcal{E}$-{succinct}-bribery is in P, then that implies that $\mathcal{E}$-bribery is also in P.
- **Encoding-based**: this refers to whether weights/prices are encoded in binary or unary.

Next are the definitions of different bribery problems corresponding to the settings above.

Definitions and Notation

Definition 4.5.1. *An instance of a bribery problem is a 4-tuple $E = (C, V, c, k)$, where*

1. *C is a list of candidates,*
2. *V is a list of voters (see below),*
3. *$c \in C$ is the candidate whom a bribing party wants to make a winner (or a unique winner in case of unique winner settings), and*
4. *k a nonnegative integer that is the bribing limit. This limit is the maximum number of voters that can be bribed or the amount of money that can be spent on bribing, i.e., the budget.*

In the bribery setting, each voter is a 3-tuple $(\mathit{prefs}, \pi, \omega)$, where

1. *prefs* is the preference of the voter (whether a list or a vector of scores depending on the election system).
2. π is the price for changing the voter's preference, and
3. ω is the weight of the voter (applicable in weighted bribery and is the same for all voters in the unweighted version and hence is ignored in that case).

Note that according to this definition, bribes are only payments made to change preferences of voters. This means that the way in which preferences are changed is not part of the definition and is not necessarily what is generally known as bribery. For example, if the same budget is spent on a campaign in order to promote (or demote) a candidate (via media for example), and it is known that spending a specific amount of money *will* result in change of preferences, then the complexity applies to this case as well. In other words, prices are only *associated* with voters, the real life mechanism of bribery, the connotations of bribed voters trading true preferences for payments, moral failure, etc. do not apply in this computational very restricted use of the word bribery.

Definition 4.5.2 ($\mathcal{E}$-bribery)**.** *Given a set C of candidates, a collection V of voters, a distinguished candidate $c \in C$ and a nonnegative integer k.*
***Question**: Is it possible to change the preference lists of at most k voters such that, under election system $\mathcal{E}$, c is a winner of the election $\mathcal{E}(C, V)$?*

Example ($\mathcal{E} = Plurality$):
Name: plurality-weighted-$bribery.
Given: A set C of candidates, a collection V of voters specified via their preference lists $(prefs_1, \ldots, prefs_m)$, their nonnegative integer weights $(w_1, \ldots, w_m)$ and their nonnegative integer prices $(p_1, \ldots, p_m)$, a distinguished candidate $c \in C$ and a nonnegative integer k (which will sometimes be referred to as *the budget*).
Question: Is there a set $B \subseteq \{1, \ldots, m\}$ such that $\Sigma_{i \in B} p_i \leq k$ and there is a way to bribe the voters from B in such a way that c becomes a winner?

Definition 4.5.3. $\mathcal{E}$-negative-bribery *is defined to be the same as $\mathcal{E}$-bribery except with the restriction that voters are not bribed to explicitly vote for the designated candidate (e.g., in plurality, this amounts to ranking that candidate first in the vote) but they are bribed in such a way that takes away the votes from other candidates.*

Definition 4.5.4 ($\mathcal{E}$-bribery$'$)**.** $\mathcal{E}$-bribery$'$ *is the basic bribery problem with the restriction that the change is not made to the whole vote but only to a unit entry in the vote, and the unit price is associaed with changing only a unit part of the vote. The exact meaning of unit change depends on the voting system, for example in approval voting, this is the change of an entry in the vote vector, while in Dodgson voting system it is a single switch of two adjecnt candidates in the preference order.*

In general,

- $\mathcal{E}$-bribery refers to the basic bribery problem for voting system $\mathcal{E}$,
- $\mathcal{E}$-weighted-bribery refers to the weighted version of the bribery problem for voting system $\mathcal{E}$,
- $\mathcal{E}$-$bribery refers to the *priced* version of the bribery problem where by *priced* we mean that each voter has a price in exchange of altering his preference, and
- $\mathcal{E}$-weighted-$bribery refers to the weighted priced version of the bribery problem (i.e., both weights and prices are considered) for voting system $\mathcal{E}$.

These are the basic flavors of the bribery problem, they can be applied to any voting system. Others are specific to particular voting systems.

4.5.3 The Computational Complexity of Bribery

The following results, for various positional scoring protocols, are all due to Faliszewski, Hemaspaandra, and Hemaspaandra [27] (full version [28]):

Results for Plurality Voting

Theorem 4.5.1. Plurality-bribery *is in* P.

Proof. The algorithm solving the basic bribery problem for Plurality is as follows: If c is not the winner then continue to bribe the supporters of the current winner to vote for c until c becomes the winner. This algorithm will terminate when c becomes the winner or when the bribing party exhausts the budget and c is not the winner. In either case, the algorithm answers the question of whether it is possible to bribe at most k voters to make c the winner–in polynomial time. □

Theorem 4.5.2. Plurality-weighted-bribery *is in* P *and* Plurality-\$bribery *is in* P*.*

Proof. The proof describes a more involved polynomial time greedy algorithm (than the one in the previous proof). The reader is referred to [28] for the full proof. □

Theorem 4.5.3. Plurality-weighted-\$bribery *is* NP*-complete, even for just two candidates.*

Remark: Neither weights alone nor the prices alone make the bribery problem hard. It is the combination of both that raises the complexity of bribery for plurality from P to NP-complete. However, the problem falls back in P if either prices or weights are encoded in unary.

Theorem 4.5.4. *Both* Plurality-weighted-\$bribery$_{unary}$ *and* Plurality-weighted$_{unary}$-\$bribery *are in* P*.*

Theorem 4.5.5. Plurality-weighted-negative-bribery *is* NP*-complete but* Plurality-negative-\$bribery *is in* P*.*

Remark: Except for negative-bribery, Plurality-weighted-bribery and Plurality-\$bribery have the same complexity, where the exact complexity depends on the particular encoding of weights/prices. In the case of negative-bribery, however, the two problems have different complexities.

Results for Approval Voting

Theorem 4.5.6. Approval-bribery *is* NP*-complete.*

Theorem 4.5.7. Approval-weighted-\$bribery′ *is* NP*-complete.*

Theorem 4.5.8. *Both* Approval-weighted-\$bribery$'_{unary}$ *and* Approval-weighted$_{unary}$-\$bribery′ *are in* P*.*

Results for Veto

Theorem 4.5.9. Veto-bribery *is in* P*.*

Proof. Given the winner as the candidate with the least number of vetoes, the algorithm that decides whether a candidate c can be made the winner by bribing at most k voters is as follows: continue on bribing voters that veto c to veto a candidate that has the least number of vetoes, until c has the least number of vetoes or the budget k is exhausted without c becoming the least vetoed. Clearly, in the former case the answer to the question in the basic bribery problem is yes, and in the latter the answer is no. The algorithm clearly runs in polynomial time. □

Results for Positinal Scoring Protocols

Theorem 4.5.10. *For every scoring protocol* $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$*,* α-{succinct}-bribery *is in* P*.*

Table 4.6 shows the complexity of each of the five bribery problems for each scoring protocol represented by the vector $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$.

Table 4.6: Dichotomy result for positinal scoring protocols.

	Scoring Protocol $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$		
Bribery Problem	$\alpha_1 = \ldots = \alpha_m$	$\alpha_1 > \alpha_2$ and $\alpha_2 = \ldots = \alpha_m$	not true that $\alpha_2 = \ldots = \alpha_m$
(Basic) $\vec{\alpha}$-bribery	P	P	P
$\vec{\alpha}$-\$bribery	P	P	P
$\vec{\alpha}$-weighted$_{unary}$-\$bribery	P	P	P
$\vec{\alpha}$-weighted-bribery	P	P	NP-complete
$\vec{\alpha}$-weighted-\$bribery	P	NP-complete	NP-complete

Results for Dodgson's, Kemeny's and Young's Systems

The following results are all due to Faliszewski et al. [28]:

Theorem 4.5.11. *For each fixed number of candidates,* DodgsonScore-{succinct}-bribery *is in* P *when restricted to that number of candidates.*

Theorem 4.5.12. *For each fixed number of candidates,* YoungScore-{succinct}-bribery *is in* P *when restricted to that number of candidates.*

Theorem 4.5.13. *For each fixed number of candidates,* Dodgson-bribery*,* Dodgson-\$bribery*,* Young-bribery*,* Young-\$bribery *are all in* P*.*

Theorem 4.5.14. *For each fixed number of candidates,* Kemeny-{succinct}-bribery *is in* P *when restricted to that number of candidates.*

A useful tool for obtaining the results above for fixed number of candidates is Lenstra's result[31] showing that the integer programming feasibility problem is in P when the number of variables is bounded (which is the case here since the number of variables is fixed). The following results are all due to Faliszewski et al. [29]:

Results for Copeland

Theorem 4.5.15. *Copeland voting system is resistant to bribery (i.e., the basic bribery problem is* NP*-hard) in both the rational voters case and the irrational voters case.*

Theorem 4.5.16. *Copeland voting system is vulnerable to* bribery$'$ *in the irrational voters case.*

[31]Presented in: H. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

Results for Llull's Voting System

Theorem 4.5.17. *Llull's voting system is resistant to bribery (i.e., the basic bribery problem is* NP*-hard) in both the rational voters case and the irrational voters case.*

Theorem 4.5.18. *Llull's voting system is vulnerable to* bribery′ *in the irrational voters case.*

Remark 1: Copeland and Llull's voting system exhibit identical complexity of bribery. Recall that both systems have very similar, almost identical, definitions except for tie-breaking rules. Therefore, tie-breaking rules do not affect the complexity of bribery for these problems (and the same results apply for the unique and multiple winner cases too).
Remark 2: The previous theorems for Copeland and Llull voting systems apply to both constructive and destructive bribery problems (which—the constructive and destructive nature—is defined analogously to constructive and destructive cases in control). However, techniques used for each case vary in nature and intricacy. The destructive case follows via greedy algorithms while for the constructive case, Copeland and Llull's election systems are modeled as a network flow problems, and that model is then used to derive results [29].

The Link between Manipulation and Bribery

Although the manipulation problem and the bribery problem are closely related, there exists voting systems where one problem is (computaionally) easy but the other is difficult. For example, this is the case with approval voting.

Theorem 4.5.19. Approval-bribery *is* NP*-complete, but* Approval-manipulation[32] *is in* P*.*

Trying to explain this result, the intuition may be something like this: since bribery involves both deciding which voters to bribe and then how to manipulate their votes to achieve a specific outcome, then bribery is a more general version of manipulation since we can think of the latter as a special case of the former where the set of voters to be influenced is known and hence it is more difficult than manipulation. Surprisingly, however, there exists a voting system where bribery is in P but manipulation is NP-complete! But it is worth to note that this system is an artificial system designed to answer the question: does there exist a voting system $\mathcal{E}$ such that the bribery is easy but manipulation is difficult? The details of the design are given in [28].

Theorem 4.5.20. *There exists a voting system $\mathcal{E}$ for which manipulation is* NP*-complete, but bribery is in* P*.*

To establish the link between bribery and manipulations, researchers have observed that to check whether bribery can be successful, it is sufficient to try all possible manipulations by k voters. If at least one is successful then we conclude that bribery can be effectuated. But this idea of trying all possible manipulations by k voters and accepting if any of these is successful resembles, in essence, the way truth-table reduction works. In light of that, the following theorem was obtained:

[32]Recall that Approval-weighted-manipulation, which is more involved than basic manipulation, is also in P whereas the basic problem of bribery for the same voting system is NP-complete.

Theorem 4.5.21. *For each fixed k, a non-priced bribery problem $\mathcal{B}$ and the analogous manipulation problem $\mathcal{M}$, it holds that $\mathcal{B} \leq^p_{dtt} \mathcal{M}$.*

The implication of this reduction is that many results obtained from the vast literature on manipulation can be used to obtain similar results for the corresponding bribery problem. The other observation—that establishes the link between manipulation and priced bribery which the previous theorem does not apply to—is that manipulation is a special case of $bribery where all manipulators have price 0, each manipulator has price 1, and the budget is equal to 0.

Theorem 4.5.22. *Let $\mathcal{E}$-$\mathcal{M}$ be some manipulation problem for voting system $\mathcal{E}$ and let $\mathcal{E}$-$\mathcal{B}$ be the analogous* $bribery *problem, it holds that $\mathcal{M} \leq^p_m \mathcal{B}$.*

The observant reader may notice that link is still missing for some bribery problem, involving weight only. The link exists indeed and the description of its more elaborated establishment can be found in [28].

4.5.4 Comments and Bibliographic Notes

- Electoral control and manipulation has been studied extensively in both social choice theory and computational social choice. The issue of bribery was only recently addressed by Faliszewski et al. [27, 29]. This has been pointed out in the work of Faliszewski et al. and it is noted again here.
- As this survey is being developed, a new work on bribery is also being developed. It extends the previous work by examining approximation algorithms for finding a bribery strategy for some well-known voting systems, including plurality [7].

4.6 Complexity of Vote Elicitation (Revisited)

We have discussed the topic of vote elicitation in Chapter 3 of this thesis. Here, we revisit some complexity-theoretic results and present them in more detail.

Recall the following definitions from Chapter 3.

Definition 4.6.1 (Effective-Elicitation)**. Given**: *A voting rule $\mathcal{E}$, a set of votes V and a number k.*
Question: *Is there a subset of V of size $\leq k$ that decides the outcome of the $\mathcal{E}$-election?*

Definition 4.6.2 (Elicitation-Not-Done)**. Given**: *A voting rule $\mathcal{E}$, a set of votes V, a number k of votes that are still unknown and a candidate c.*
Question: *Is there a way to cast the k votes so that c will not win the $\mathcal{E}$-election?*

Definition 4.6.3 (Possible and Necessary Winners)**.** *Let $\mathcal{E}$ be a voting rule, X a set of alternatives, and R^n a collection of (possibly incomplete) preference profiles.*

- *$x \in X$ is a necessary winner for R^n with respect to $\mathcal{E}$ if and only if for all $\mathcal{T} \in Ext(R^n)$, $x \in \mathcal{E}(\mathcal{T})$. The set of necessary winners under $\mathcal{E}$ given R^n is denoted by $NW_{\mathcal{E}}(R^n)$.*
- *$x \in X$ is a possible winner for R^n with respect to $\mathcal{E}$ if and only if there exists a $\mathcal{T} \in Ext(R^n)$ such that $x \in \mathcal{E}(\mathcal{T})$. The set of possible winners under $\mathcal{E}$ given R^n is denoted by $PW_{\mathcal{E}}(R^n)$.*

Now after establishing foundational results in this chapter, we revisit selected theorems and present their proofs. The theorems are stated (but proofs of only some are given) in cited work. Here, some of those proofs are provided with more explanation:

Theorem 4.6.1. *For the Approval protocol,* Effective-Elicitation *is* NP*-complete [14].*

Proof. First, given the unknown votes, it can be verified in polynomial time whether the outcome of the election coincides with the preferences of the voters in the subset of V of size $\leq k$.

Second, to show completeness, the proof proceeds as a reduction from 3-Cover[33]. Reduce an instance of X3C to the following instance of *Effective-Elicitation*: Let the candidate set be $X \cup \{w\}$. There are $2r - 2q + 2$ voters voting as follows: for every subset S_i in the collection of r subsets of X, there is a vote approving $S_i \cup w$. Additionally, there are $r - 2q + 2$ votes approving only w. Finally, let $k = r - q + 2$.

To show that these problems instances are equivalent, first suppose that there is a X3C, then we elicit all the votes that approve only w and the votes that correspond to sets in the cover . The total is $r - 2q + 2 + q = r - q + 2 = k$ votes. Then w gains $r - q + 1$ more points than all other candidates (there are only $r - q$ votes remaining to be distributed among all of the remaining candidates). Hence there is an effective elicitation. The other direction can be shown by a contrapositive argument. Suppose there is no 3-Cover, then

[33] 3-Cover is an NP-complete problem which is defined in the first section of this chapter.

eliciting k votes will always give one of the candidates in X at least 2 votes. This means w can be at most $r - q$ points ahead of this candidate which is not enough to guarantee winning the overall election. Therefore, with $r - q$ votes remaining, the outcome cannot be decided. So there is no effective elicitation. □

Theorem 4.6.2. *For STV, deciding whether $c \in C$ is a possible winner is* NP*-complete [53].*

Proof. First, to show that the problem is in NP, note that given a completion of the preference profiles collection, it can be checked in polynomial time whether c wins.

Second, hardness follows from the result that Effective-Preference for STV is NP-complete [3]. □

The following table summerizes the results on the complexity of vote elicitation and related problems. Note that P is a subset of NP by definition, we therefore do not emphesize that a problem which is in P is also in NP.

Table 4.7: Summary of results on vote elicitation and related problems.

Problem	**Voting Rule**	**Complexity**	**Result in**
Effective-Elicitation	STV	NP-complete	[14]
Effective-Elicitation	Borda	NP-complete	[14]
Effective-Elicitation	Copeland	NP-complete	[14]
Effective-Elicitation	Maximin	NP-complete	[14]
Necessary Winners	Condorcet	P	[42]
Necessary Winners	Scoring Protocols	P	[42]
Necessary Winners	STV	coNP-complete	[53]
Possible Winners	Scoring Protocols	P	[42]
Possible Winners	STV	NP-complete	[53]
Possible Winners	Condorcet	P	[42]
Is $c \in C$ a Necessary Winner?	Any $\mathcal{E} \in$ P	coNP	[42]
Is $c \in C$ a Necessary Winner?	General $\mathcal{E}$	coNP-hard	[53]
Is $c \in C$ a Possible Winner?	Any $\mathcal{E} \in$ P	NP	[42]
Is $c \in C$ a Possible Winner?	General $\mathcal{E}$	NP	[53]
Determining winners via Vote Elicitation	Any $\mathcal{E} \in$ P satisfying IIA	P	[53]
Finding $PW^* \subseteq PW$ s.t. $\|PW^*\| < k\|PW\|$, $k \in \mathbb{Z}^+$	STV	NP-hard	[53]
Manipulation	Any $\mathcal{E}$ s.t. PW, NW $\in$ P	P	[42]

Key: NW = Necessary Winner, PW = Possible Winner.

4.7 The Complexity of Communication in Voting Rules

4.7.1 What is Communication Complexity?

The first basic model of a communication problems is due to Yao [64]. In their overview of communication complexity [43], Kushilevits and Nisan describe in simple terms the communication problems as follows:

"A system must perform some task that depends on information distributed among different parts of the system (called *processors, parties,* or *players*). The players thus need to communicate with each other in order to perform the task" [43].

In the context of voting rules, the system mentioned in Kushilevits and Nisan's description above is a given voting rule, the players are the voters and the communication needs to take place between the elicitor protocol and the voters.

The question that arises in the context of communication complexity of voting is the following: assuming nothing is known about voters' preferences a priori, what is the worst-case number of bits that needs to be communicated to execute a voting rule and find the outcome? The answer to this question for a given voting rule is the *communication complexity* of that rule. In general, the (deterministic) communication complexity of a problem is the worst-case number of bits communicated in the best (deterministic) algorithm for solving the problem.

It is important to make the distinction between a voting rule and a communication protocol clear. The voting rule maps votes into outcomes (whether full ranking or winner set), independently of how these votes are communicated or how the preferences are elicited. The latter is the role of a communication protocol, it actually determinates (relevant parts of the) votes by eliciting relevant information about preferences from the voters.

4.7.2 Communication Complexity of Voting Rules

The situation of communication in voting rules can be detailed as follows. Each voter i, where $i \in \{1, \ldots, n\}$ knows its vote v_i, the goal is to compute the outcome of the voting rule $\mathcal{E}$ based on the votes of all voters, $\mathcal{E}(v_1, v_2, \ldots, v_n)$. In each stage of the process, a voter discloses a *bit* of information based on its vote and the bits communicated so far. Ultimately, the outcome of $\mathcal{E}(v_1, v_2, \ldots, v_n)$ is determined and becomes known to all voters. This is described as a *deterministic* communication protocol.

In a nondeterministic protocol, the next bit to sent is (possibly) chosen nondeterministically.

The aim, in either cases of determinism and nondeterminism, is to minimize the worst-case number of bits communicated.

4.7.3 The Importance of Communication Complexity

It is argued that when elicitation is "clever," in other words, when the protocol asks minimum number of queries to the right agents in order to determine the outcome; it can reduce the communication overhead and also make the whole process more efficient. This is because preference aggregation can be infeasible when the total number of preferences (votes) is too large to communicate or to consider. From an agent-perspective, and especially in software multi-agent systems, it can also reduce the communication burden on the agent which needs its computational resources to gather information and determine preferences before communicating them. It is also argued that reduced communication requirements “preserve (some of) the agents’ privacy” [17], although this advantage can also be seen as a disadvantage since agents can exploit the communication protocol to effect strategic voting in favor of individual interests.

4.7.4 Communicating Votes and Strategic Voting

The choice of the communication protocol may further influence the strategic behavior of the voters. For example, a multistage protocol, such as plurality with runoff, reveals through the queries asked in the second round, the top two candidates of the previous round before the outcome is actually declared. This indicates the close relation between the amount of communication that takes place in the elicitation process and the opportunities created for strategic voting, which again reminds us of the close connection between vote elicitation and the manipulation problem.

4.7.5 Results

In this section, we present the results of work related to communication complexity of voting rules [60, 17]. The first work, however, applies mostly to intersection-monotonic rules. Intersection monotonicity is a strong form of monotonicity[34]. For rules with this property, such as Approval and Condorcet, both work contribute similar results and can be referred to for alternative proofs.

Given m as the number of candidates and n as the number of voters, the following is a list of results pertaining to the communication complexity of common voting rules. All these results—except for Theorem 4.7.9 which is proven independently here—are due to authors as indicated in the citation of each result. The proofs presented here, however based on proofs in cited work, are rewritten here and expanded with more detail.

Theorem 4.7.1. *The deterministic communication complexity of any ranked-based voting rule is $O(nm\, log\, m)$ [17].*

Proof. First, reporting the rank of a given candidate c requires $O(log\, m)$ bits per voter. This is because each voter knows its ranking of alternatives and there are m alternatives, the position of any can be represented by $O(log\, m)$ bits.
Second, let every voter communicates his entire ranking which requires, for each voter, $O(m\, log\, m)$ bits, then the total rankings of n voters require $O(nm\, log\, m)$ communicated bits. □

Theorem 4.7.2. *The deterministic communication complexity of Plurality is $O(n\, log\, m)$ [17].*

[34]Monotonicity and other properites of voting rules are defined in Chapter 1 of this thesis.

Proof. As above, indicating the rank of a given candidate requires $O(log\, m)$ bits. Since the plurality rule only considers first choice candidates, every voter communicate her first choice candidate only, and since there are n voters, the deterministic communication complexity of the plurality rule is $O(n\, log\, m)$. □

Theorem 4.7.3. *The nondeterministic communication complexity of Plurality is $\Omega(n\, log\, m)$ (even to decide whether a given candidate has won [35]) [17].*

Theorem 4.7.4. *The deterministic communication complexity of Plurality with Runoff is $O(n\, log\, m)$ [17].*

Proof. An argument similar to the one in the proof of deterministic communication complexity of Plurality can be used to show that the communication complexity of the first stage of plurality with runoff is $O(n\, log\, m)$.
In the second stage, each voter communicates the preferred candidate among the top two which requires a single bit for every voter and since there are n voters the total communication requirement is $O(n)$ bits.
The overall deterministic communication complexity of Plurality with Runoff is therefore $O(n\, log\, m)$. □

Theorem 4.7.5. *The nondeterministic communication complexity of Plurality with Runoff is $\Omega(n\, log\, m)$ (even to decide whether a given candidate has won) [17].*

Theorem 4.7.6. *The deterministic communication complexity of STV is $O(n(log\, m)^2)$ [17].*

Theorem 4.7.7. *The nondeterministic communication complexity of STV [36] is $\Omega(n\, log\, m)$ [17].*

Theorem 4.7.8. *The nondeterministic communication complexity of each of Borda rule and the Copeland rule is $\Omega(nm\, log\, m)$ (even to decide whether a given candidate has won).*

Theorem 4.7.9. *The deterministic communication complexity of the Borda rule is $O(nm\, log\, m)$.*

Proof. In the Borda count, each voter ranks all alternatives, and alternatives are given points based on their position in the vote. Finally alternatives are ranked based on their total number of points.
Since ranking an alternative requires $O(log\, m)$ bits and there are m candidates, each voter requires $O(m\, log\, m)$ bits to communicate the vote, and since there are n voters, the total number of bits communicated is $O(nm\, log\, m)$. □

Theorem 4.7.10. *The nondeterministic communication complexity of Maximin is $\Theta(mn)$.*

Proof. Directly from the definition of the tight bound Θ and related upper and lower bounds obtained in [17]. □

[35] Although this lower bound is the same as the upper bound in the deterministic case, the proof is much more involved and the reader is referred to the brief version of the proof in [17].

[36] The proof was omitted in the cited paper.

Theorem 4.7.11. *The deterministic communication complexity of each of the Condorcet rule, Approval voting, Cup, and Bucklin rule is* $O(nm)$ *[17].*

Theorem 4.7.12. *The nondeterministic communication complexity of the Condorcet rule is* $\Omega(nm)$ *[60, 17].*

Theorem 4.7.13. *The nondeterministic communication complexity of each of the Approval voting, Cup, and Bucklin rule is* $\Omega(nm)$ *[17].*

The following table summarizes the results obtained. All upper bound are deterministic except for Maximin, the entry for Maximin under the upper bound is nondeterministic, the best deterministic upper bound found for Maximim communication complexity in [17] is $O(nm \, log \, m)$.

Table 4.8: Communication complexity of voting rules sorted from low to high.

Voting Rule	**Nondeterministic Lower Bound**	**(Deterministic) Upper Bound**
Plurality	$\Omega(n \, log \, m)$	$O(n \, log \, m)$
Plurality with runoff	$\Omega(n \, log \, m)$	$O(n \, log \, m)$
STV	$\Omega(n \, log \, m)$	$O(n(log \, m)^2)$
Condorcet	$\Omega(nm)$	$O(nm)$
Approval	$\Omega(nm)$	$O(nm)$
Bucklin	$\Omega(nm)$	$O(nm)$
Cup	$\Omega(nm)$	$O(nm)$
Maximin	$\Omega(nm)$	$O(nm)$
Borda	$\Omega(nm \, log \, m)$	$O(nm \, log \, m)$
Copeland	$\Omega(nm \, log \, m)$	$O(nm \, log \, m)$

4.8 New Criteria for Evaluating Voting Rules

Chapter 1 stated a list of properties of voting rules. Those properties, such as Anonymity, IIA, and Pareto Optimality, were proposed by social choice theorists based on the axiomatic (and sometimes practical) analysis of voting rules. In this chapter, we have discussed six main problems studied in the literature of complexity voting theory. Based on these discussions and related results, we present a new set of (computational) criteria for evaluating voting rules:

- Efficient winner determination process (i.e., polynomial time winner problem).
- Resistance against control by the authority conducting the election (i.e., NP-hardness of control).
- Resistance against manipulation by voters (i.e., NP-hardness of manipulation).
- Resistance against bribery (i.e., NP-hardness of bribery).
- Efficient/Intractable vote elicitation process (depending on the purpose of vote elicitation, i.e., whether it is for efficiently finding winners or for realizing results by manipulation).
- Efficient vote communication process (i.e., polynomial time vote communication).

Chapter 4 Bibliography

[1] Ailon, N., Charikar, M., and Newman, A. Aggregating inconsistent information: Ranking and clustering. In *STOC '05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 2005), ACM Press, pp. 684–693.

[2] Banks, J. Sophisticated voting outcomes and agenda control. Working Papers 524, Division of the Humanities and Social Sciences, California Institute of Technology, 1984.

[3] Bartholdi, III, J., and Orlin, J. Single transferable vote resists strategic voting. *Social Choice and Welfare 8*, 4 (1991), 341–354.

[4] Bartholdi, III, J., Tovey, C., and Trick, M. The computational difficulty of manipulating an election. *Social Choice and Welfare 6* (1989), 227–241.

[5] Bartholdi, III, J., Tovey, C., and Trick, M. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare 6* (1989), 157–165.

[6] Bartholdi, III, J., Tovey, C., and Trick, M. How hard is it to control an election? *Mathematical and Computer Modelling 16* (1992), 27–40.

[7] Brelsford, E. Approximation and elections. Master's thesis, Computer Science Department, Rochester Institute of Technology, 2007.

[8] Chamberlin, J., and Courant, P. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review 77*, 4 (1983), 718–733.

[9] Coleman, T., and Teague, V. On the complexity of manipulating elections. In *CATS' 07: Proceedings of the 13th Computing Conference: The Australasian Theory Symposium* (Ballarat, Australia, 2007), J. Gudmundsson and B. Jay, Eds., vol. 65 of *CRPIT*, ACS, pp. 25–33.

[10] Conitzer, V. Computing Slater rankings using similarities among candidates. In *AAAI' 06: Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference* (Boston, MA, USA, 2006), AAAI Press.

[11] Conitzer, V., Davenport, A., and Kalagnanam, J. Improved bounds for computing Kemeny rankings. In *AAAI' 06: Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference* (Boston, MA, USA, 2006), AAAI Press.

[12] Conitzer, V., Lang, J., and Sandholm, T. How many candidates are needed to make elections hard to manipulate? In *TARK '03: Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge* (New York, NY, USA, 2003), ACM Press, pp. 201–214.

[13] Conitzer, V., and Sandholm, T. Complexity of manipulating elections with few candidates. In *AAAI' 02: Proceedings of the 18th National Conference on Artificial Intelligence* (Menlo Park, CA, USA, 2002), AAAI Press, pp. 314–319.

[14] Conitzer, V., and Sandholm, T. Vote elicitation: Complexity and strategy-proofness. In *AAAI' 02: Proceedings of the 18th National Conference on Artificial Intelligence* (Menlo Park, CA, USA, 2002), AAAI Press, pp. 392–397.

[15] Conitzer, V., and Sandholm, T. Universal voting protocol tweaks to make manipulation hard. In *IJCAI' 03: Proceedings of the 18th International Joint Conference on Artificial Intelligence* (Acapulco, Mexico, 2003), pp. 781–788.

[16] Conitzer, V., and Sandholm, T. Common voting rules as maximum likelihood estimators. In *UAI' 05: Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence* (Edinburgh, Scotland, UK, 2005), pp. 145–152.

[17] Conitzer, V., and Sandholm, T. Communication complexity of common voting rules. In *EC '05: Proceedings of the 6th ACM Conference on Electronic Commerce* (New York, NY, USA, 2005), ACM Press, pp. 78–87.

[18] Conitzer, V., and Sandholm, T. Nonexistence of voting rules that are usually hard to manipulate. In *AAAI' 06: Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Conference on Innovative Applications of Artificial Intelligence* (Boston, MA, USA, 2006), AAAI Press.

[19] Conitzer, V., Sandholm, T., and Lang, J. When are elections with few candidates hard to manipulate? *Journal of the ACM 54*, 3 (2007), 14.

[20] Cormen, T., Leiserson, C., Rivest, R., and Stein, C. *Introduction to Algorithms*. MIT Press, Cambridge, MA, USA, 2001.

[21] Davenport, A., and Kalagnanam, J. A computational study of the Kemeny rule for preference aggregation. In *AAAI' 04: Proceedings of the 19th National Conference on Artificial Intelligence and the 16th Conference on Innovative Applications of Artificial Intelligence* (San Jose, CA, USA, 2004), AAAI Press / The MIT Press, pp. 697–702.

[22] Diaconis, P., and Graham, R. Spearman's footrule as a measure of disarray. Technical Report ADA030697, California Department Statistics, Stanford University, 1976.

[23] Dummett, M., and Farquharson, R. Stability in voting. *Econometrica 29*, 1 (1961), 33–43.

[24] Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. Rank aggregation methods for the Web. In *World Wide Web* (2001), pp. 613–622.

[25] Elkind, E., and Lipmaa, H. Hybrid voting protocols and hardness of manipulation. In *ISAAC' 05: Proceedings of the 16th International Symposium on Algorithms and Computation* (Sanya, Hainan, China, 2005), X. Deng and D.-Z. Du, Eds., vol. 3827 of *Lecture Notes in Computer Science*, Springer, pp. 206–215.

[26] Elkind, E., and Lipmaa, H. Small coalitions cannot manipulate voting. In *FC' 05: Proceedings of the 9th International Conference on Financial Cryptography and Data Security* (2005), A. Patrick and M. Yung, Eds., vol. 3570 of *Lecture Notes in Computer Science*, Springer, pp. 285–297.

[27] Faliszewski, P., Hemaspaandra, E., and Hemaspaandra, L. The complexity of bribery in elections. In *AAAI' 06: Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Conference on Innovative Applications of Artificial Intelligence* (Boston, MA, USA, 2006), AAAI Press.

[28] Faliszewski, P., Hemaspaandra, E., and Hemaspaandra, L. How hard is bribery in elections? Technical Report TR-895, Department of Computer Science, University of Rochester, 2006.

[29] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., and Rothe, J. Llull and Copeland voting broadly resist bribery and control. In *AAAI' 07: Proceedings of the 22nd National Conference on Artificial Intelligence* (Vancouver, Canada, 2007), pp. 724–730.

[30] Fishburn, P. Condorcet social choice functions. *SIAM Journal on Applied Mathematics 33*, 3 (1977), 469–489.

[31] Garey, M., and Johnson, D. *Computers and Intractability; A Guide to the Theory of* NP*-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[32] Gibbard, A. Manipulation of voting schemes: A general result. *Econometrica 41*, 1 (1973), 587–601.

[33] Hemaspaandra, E., and Hemaspaandra, L. Dichotomy for voting systems. Technical Report TR-861, Department of Computer Science, University of Rochester, 2005.

[34] Hemaspaandra, E., and Hemaspaandra, L. Dichotomy for voting systems. *Journal of Computer and System Sciences 73*, 1 (2007), 73–83.

[35] Hemaspaandra, E., Hemaspaandra, L., and Rothe, J. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM 44*, 6 (1997), 806–825.

[36] Hemaspaandra, E., Hemaspaandra, L., and Rothe, J. Anyone but him: The complexity of precluding an alternative. In *AAAI' 05: Proceedings of the 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference* (Pittsburgh, PE, USA, 2005), AAAI Press / The MIT Press, pp. 95–101.

[37] Hemaspaandra, E., Hemaspaandra, L., and Rothe, J. Hybrid elections broaden complexity-theoretic resistance to control. Technical Report TR-900, Department of Computer Science, University of Rochester, 2006. Conference version appears in *IJCAI' 07: Proceedings of the 20th International Joint Conference on Artificial Intelligence* (Hyderabad, India, 2007).

[38] Hemaspaandra, E., Spakowski, H., and Vogel, J. The complexity of Kemeny elections. *Theoretical Computer Science 349*, 3 (2005), 382–391.

[39] Hemaspaandra, L., and Ogihara, M. *The Complexity Theory Companion.* Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[40] Homan, C., and Hemaspaandra, L. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. In *MFCS' 06: Proceedings of the Mathematical Foundations of Computer Science* (Stará Lesná, Slovakia, 2006), vol. 4162 of *Lecture Notes in Computer Science*, Springer, pp. 528–539.

[41] Hopcroft, J., Motwani, R., and Ullman, J. *Introduction to Automata Theory, Languages, and Computation*, second ed. Addison-Wesley, 2001.

[42] Konczak, K., and Lang, J. Voting procedures with incomplete preferences. In *Proceedings of the IJCAI' 05 Multidisciplinary Workshop on Advances in Preference Handling* (Edinburg, Scotland, 2005).

[43] Kushilevitz, E., and Nisan, N. *Communication Complexity.* Cambridge University Press, 1996.

[44] Lang, J., Pini, M., Rossi, F., Venable, K., and Walsh, T. Winner determination in sequential majority voting. In *IJCAI' 07: Proceedings of the 20th International Joint Conference on Artificial Intelligence* (Hyderabad, India, 2007), pp. 1372–1377.

[45] Linz, P. *An Introduction to Formal Languages and Automata*, fourth ed. Jones and Bartlett, Sudbury, MA, USA, 2006.

[46] Marquis de Condorcet. Essai sur l'application de l'analyse à la probabilité des decisions rendues à la pluralite des voix, 1785.

[47] Martin, J. *Introduction to Languages and the Theory of Computation*, third ed. McGraw-Hill, 2002.

[48] Maus, S., Peters, H., and Storcken, T. Anonymous voting and minimal manipulability. Research Memoranda 009, Maastricht: METEOR, Maastricht Research School of Economics of Technology and Organization, 2005. Available at `http://ideas.repec.org/p/dgr/umamet/2005009.html`.

[49] Monroe, B. Fully proportional representation. *American Political Science Review 89*, 4 (1995), 925–940.

[50] Moulin, H. *Axioms of Cooperative Decision Making*, second ed. Econometric Society Monographs. Cambridge University Press, 2004.

[51] Pini, M., Rossi, F., Venable, K., and Walsh, T. Aggregating partially ordered preferences: Impossibility and possibility results. In *TARK '05: Proceedings of the 10th Conference on Theoretical Aspects of Rationality and Knowledge* (Singapore, Singapore, 2005), National University of Singapore, pp. 193–206.

[52] Pini, M., Rossi, F., Venable, K., and Walsh, T. Strategic voting when aggregating partially ordered preferences. In *AAMAS '06: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems* (New York, NY, USA, 2006), ACM Press, pp. 685–687.

[53] Pini, M., Rossi, F., Venable, K., and Walsh, T. Incompleteness and incomparability in preference aggregation. In *Proceedings of the IJCAI' 07 Multidisciplinary Workshop on Advances in Preference Handling* (2007), pp. 1464–1469.

[54] Pritchard, G., and Slinko, A. On the average minimum size of a manipulating coalition. *Social Choice and Welfare 27*, 2 (2006), 263–277. Available at `http://ideas.repec.org/a/spr/sochwe/v27y2006i2p263-277.html`.

[55] Procaccia, A., and Rosenschein, J. Junta distributions and the average-case complexity of manipulating elections. In *AAMAS '06: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems* (New York, NY, USA, 2006), ACM Press, pp. 497–504.

[56] Procaccia, A., Rosenschein, J., and Zohar, A. Multi-winner elections: Complexity of manipulation, control and winner-determination. In *IJCAI' 07: Proceedings of the 20th International Joint Conference on Artificial Intelligence* (Hyderabad, India, 2007), pp. 1476–1481.

[57] Rossi, F., Venable, K., and Walsh, T. Aggregating preferences cannot be fair. *Intelligenza Artificiale 2*, 1 (2005), 30–38.

[58] Rothe, J., Spakowski, H., and Vogel, J. Exact complexity of the winner problem for Young elections. *Theory of Computing Systems 36*, 4 (2003), 375–386.

[59] Satterthwaite, M. A. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory 10*, 1 (1974), 187–217.

[60] Segal, I. The communication requirements of of social choice rules and supporting budget sets. Economics Working Papers 0039, Institute for Advanced Study, School of Social Science, Princeton University, 2004. Available at `http://ideas.repec.org/p/ads/wpaper/0039.html`.

[61] Vickrey, W. Utility, strategy, and social decision rules. *The Quarterly Journal of Economics 74*, 4 (1960), 507–535.

[62] Wakabayashi, Y. Aggregation of binary relations: Algorithmic and polyhedral investigations, 1986. Doctoral Dissertation. University of Augsburg.

[63] Wegener, I. *Complexity Theory: Exploring the Limits of Efficient Algorithms.* Springer, Berlin, Germany, 2005.

[64] Yao, A. Some complexity questions related to distributive computing (preliminary report). In *STOC '79: Proceedings of the 11th Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1979), ACM Press, pp. 209–213.

[65] Young, H. Optimal voting rules. *Journal of Economic Perspectives 9*, 1 (1995), 51–64.

Chapter 5

Applications of Voting in Computing

The vote is the most powerful instrument ever devised by man...

Lyndon B. Johnson

5.0 Prologue

As a general, simple and appealing method for aggregating preferences, voting has been devised for a wide range of applications spanning economics, operations research, linguistics, mathematics, computer science and other interdisciplinary fields such as information science and sociology. This is in addition, of course, to its essential importance in political science and social choice theory.

Within computer science and related disciplines, voting applications traverse areas such as: Artificial Intelligence (AI), Databases (DB), Networking, Natural Language Processing (NLP), Information Retrieval (IR) and e-Commerce. In addition, both the theoretical study of voting and its applications have also influenced areas such as Complexity Theory, Software Engineering (SE), and Cryptography and Security.

Although these are the main areas where voting applications in computing are found, these applications are not confined to this small list of areas. It seems that wherever there are preferences, there will probably be a voting scheme suggested to aggregate those preferences. And situations that can be interpreted as a group decision-making problem based on individual preferences are not uncommon in automated processing. Therefore, scattered employment of voting appears in software systems and applications serving various domains.

In this chapter, we briefly list some of voting applications in computing. The sections of the chapter provide a classification of the applications based on the computing field in which they were devised. This classification is further refined in subsections; forming a hierarchy of voting applications in computing. The order of a subject in the list reflects, to a limited extent, the prevalence of applying voting in that subject in comparison to others.

It is worth observing that this chapter is not intended to introduce the computing disciplines mentioned here, for that has different avenues and is clearly not the focus of this chapter.

5.1 Applications of Voting in Artificial Intelligence

5.1.1 Voting in Multi-agent Planning

In distributed AI, a group of autonomous agents needs to coordinate individual actions in order to accomplish a goal (as a group). At each step of processing, agents must agree on the next step to take. This is ideally a consensus or an agreement based on some preference aggregation mechanism. Negotiation has been previously used in such situations. Recently, voting has been proposed as preference aggregation mechanics for multi-agent planning [6, 7]. At each step, instead of negotiation over the next joint action, each agent votes for the next preferred action in the group plan and individual preferences are aggregated using a voting procedure. This approach enjoys an effective coordination of tasks via voting whilst keeping the process simple and natural in spirit (an appealing characteristic of a solution in AI).

5.1.2 Voting in Collaborative Filtering

In the context of Collaborative Filtering (CF), users have preferences over a domain of items and a system is to recommend a subset of items to a user based on the preferences of all users in the group. Hence, the collaboration (of users) and filtering (of items) in the method name. In CF, a prediction of a user's preferences is viewed as a function of the preferences of all users, and a recommendation is made based on this prediction. This means that some aggregation of preferences must take place to form these predictions. Voting as a natural method for aggregating preferences has been effected in collaborative filtering algorithms. Each user votes over its preferred item(s) and items are then assigned (average) scores based on the number of votes received, preferences over items are finally predicted based on items' scores. Weighted voting is suggested as an enhancement to the filtering process.

This approach to CF has many applications especially in e-commerce and information retrieval.

Besides applying voting to collaborative filtering, a theoretical axiomatization of this alternative technique of CF was presented in a framework similar to that of traditional social choice theory [22]. Through the application of voting and by drawing from the long-studied aspects of voting procedures in social choice theory, the proposal bridges the two disciplines at a more profound level and initiates the foundational analysis of CF which was previously discussed from an empirical perspective only.

5.2 Voting and Ranking Systems

In typical ranking system setting, the set of alternatives and the set of agents coincide. Each agent ranks other agents and then all agents' rankings are aggregated in order to obtain a "social" ranking. Ranking Systems have many applications in Internet technology, e-commerce and the intersection of these two. The axiomatic foundation of ranking systems is closely related to and based on that of voting theory, this line of research has been initiated to reduce the gap between ranking algorithms and the mathematical theory of social choice [2, 3]. In addition to the theoretical link, applications of voting in ranking systems are numerous. We have already discussed a side of that in the context of collaborative filtering. Another important application of voting is in ranking methods for the Web [5].

Ranking Methods for the Web

applications of voting in computing!ranking systems!ranking methods for the Web Inspired by the extended Condorcet criterion (ECC) and Kemeny's voting rule in combination with other techniques, a rank aggregation for the web was suggested to address different problems facing web-based ranking systems [5]. The ECC is both efficient and particularly effective in combating spam. The idea of Kemeny's rule is used in meta-search engines that produce results based on the rankings of several search engines by minimizing the disagreement of several input rankings, in a similar way to which Kemeny's rule minimizes the distance to a consensus ordering of the alternatives. The computational difficulty of the Kemeny's rule was managed using a "local" Kemenization process that is a relaxation of the optimal rule yet satisfying the extended Condorcet criterion and is computationally tractable.

5.3 Applications of Voting in Information Retrieval

5.3.1 Information Extraction

Information Extraction (IE) is a form of Information Retrieval where shallow text processing is performed to automatically extract structured information from unstructured documents. A structure may refer to a category (a class), a well defined contextual information or semantics associated with the extracted piece of information.

Voting is used in classifying a set of attributes or features. This is done by combining the output of multiple classifiers through voting. Each classifier is queried for the class value and the class with highest frequency is selected. In other words, the majority voting rule is used to select a class value where classifiers are voters and classes are candidates. Other voting rules have also been suggested for this purpose, for example weighted plurality, where a classifier's vote is weighted by its accuracy, and voting over class probability distributions.

Voting was compared with other methods of classification and turned to be computationally cheaper and was argued to be the "the simplest form of loosely integrating the output of multiple components" [25].

5.4 Applications of Voting in Databases and Networking

5.4.1 Data Consistency Maintenance

When data files are replicated to ensure availability, for example in case of a network communication failure or site connectivity problem, algorithms managing the several replica are needed to ensure consistency of replicated data. When a link breaks within a network, the network splits into disjoint smaller networks. Each of these networks is a connected *partition*. The sites within the same partition can communicate with each other but no site can communicate with sites from a different partition.

The files of a database may be replicated at different sites, when a network is partitioned, attempts to update replicated files by various disconnected sites may cause an inconsistency problem. One way of enforcing consistency is by allowing only one partition to update files at any given time. This distinguished partition, if any, is the partition that contains more than half of the sites, and is therefore called the *majority* partition [28, 29, 10]. Using voting this way, "sites" are voters and partitions are "candidates." The majority rule is a simple and effective rule for avoiding conflicting updates since only a unique partition can contain more than half of the sites at a time and hence only this partition can update the files. The voting approach to maintaining replicated data consistency was proposed by many researchers using various enhancements such as weighted voting and "dynamic" voting which permits updates in a partition provided it contains more than half of the *up-to-date* copies of the replicated file which arguably improves availability [16, 17, 27].

5.4.2 Similarity Search and Classification

A database of "high dimensional" data is viewed as a set of database elements where each element is a vector (of a specific dimension) in a Euclidean space. A query to this database tries to find elements identical to the specification of the query or similar elements.
A novel approach to efficient similarity search suggests voting to rank the elements based on their similarity to the query. The rankings are then combined using an aggregation algorithm [9].

In this approach, each coordinate of the n-dimensional Euclidean space is considered as a "voter" and the m database points (entries) are "candidates" in an election for candidates that are nearest to the query specification. Voter j, for $1 \leq j \leq n$, ranks the m candidates based on how close they are to the query in the j-th coordinate. This yields n rankings of the candidates, and the goal is to aggregate those rankings in a second step to produce a single ordering on the candidates.

This approach further imports concepts from voting theory to solve the problem of similarity search in high dimensional databases. Since, as was mentioned in the ranking systems application above, the Condorcet criterion is found to produce robust results that are "spam-proof." An algorithm that is based on Kemeny's rule (as a Condorcet consistent rule) is used to aggregate the individual rankings since it "minimizes" distance to a goal ranking and the problem here is to minimize differences between an element returned as a query answer and the specification of the query, equivalently achieving high similarity between queries and results.

The NP-complete complexity of the aggregation process is overcome using heuristics tailored to this specific problem and by observing that only the ranking of top candidates in the final outcome is needed and not the full ranking of all candidates.

5.4.3 Applications of Voting in Networking

A voting-based clustering algorithm for extending the lifetime of sensor networks was proposed as an energy-efficient, balanced and yet simple solution for the energy problem in sensor networks [23]. In this approach, sensors vote for their neighbors to elect suitable cluster heads.

Voting has also been devised for achieving mutual exclusion and synchronizing reading and writing of data replicated across different segments of a network [12, 1].

Moreover, aspects pertinent to traditional elections such as manipulation have also been studied for applications of elections in networking, for example see [26].

5.5 Applications of Voting in Natural Language Processing

Morphological Disambiguation of Text

Morphological disambiguation refers to assigning part-of-speech tags to words in the input text. Although the tag is usually the part-of-speech of the word, it could be any other morphological or grammatical feature depending on the application.

A constraint-based morphological disambiguation system was implemented in which individual constraints vote on matching morphological tags (parses), and disambiguation of all the tokens (words) in a sentence is performed at the end by selecting parses that receive the highest votes. The accuracy of this approach for disambiguating Turkish text reached 95% [21].

5.6 Contributions to Complexity Theory

Complexity-theoretic aspects of voting schemes and related problems have been extensively studied. One of the most recognized results of studying the computational complexity of voting is finding "natural" computationally difficult problems. Many problems in the voting literature were found to be NP-complete. These problems are more widely understood than some of the well-known basic NP-complete problems. Moreover, there are complexity classes for which no "natural" complete problem had been found, such as the class of problems solvable in polynomial time by parallel access to NP, $\mathrm{P}_{\parallel}^{\mathrm{NP}}$. The first natural problems to be found complete for parallel access to NP were the winner and ranking problems for Lewis Carroll's 1876 voting scheme and Young election system due to Hemaspaandra et al. [14] and Rothe et al. [24] respectively. The third natural problem in the complexity class $\mathrm{P}_{\parallel}^{\mathrm{NP}}$ was the winner problem for Kemeny's voting system. The result appeared in 2005 and is due to Hemaspaandra et al. [15].

5.7 Other Applications

5.7.1 Logic Programming and Voting for Scheduling

In this application, voting is suggested as a remedy to the problem of scheduling a meeting for individuals with conflicting unavailability and hence conflicting preferences over meeting dates [18]. Logic programming is used to express individual and subgroup attendee availability for given dates. Logic (answer set) programming rules are defined to generate possible meetings based on the input and finally a voting rule is encoded using a third set of rules and is integrated into the meeting-scheduler system. The voting procedure used is inspired by voting procedures defined on partial preferences [19] that are discussed in the chapter on preferences of this thesis. This is driven by the nature of the problem where a meeting date needs to be finalized based on possibly partial preferences over different days for meeting or because of the conflicting nature of full preferences when put together. This work draws clearly from previous work on computational aspects of voting yet it is the first to marry the concepts of answer set programming and that of voting.

5.7.2 Voting in Entertainment Recommender System

A movie recommender system that caters to users' interests and resolves conflicting preferences using voting has been developed and discussed in [11]. In this application of voting in recommender systems, Black's voting rule[1] was applied on movies as the set of candidates and a set of dimensions (criteria) associated with weights and values. As described in [11], the vote for a given movie on each dimension is a product of the weight on that dimension multiplied by the relative rankings of the elements within the dimension. The top few vote receivers are returned as recommended by the system. The estimated value or rating of a movie to be recommended is calculated as the ratio of the number of votes it gets to the maximum number of votes a movie can get for the given user preferences. Further, the system was argued to be effective and robust to inaccuracies and imprecision (in stated preference) reflecting the robustness of Black's rule to small variations in the user preferences compared to proportional ranking mechanism.

5.8 Electronic Voting

Perhaps the most explicit form of combining voting and computing is realized in electronic voting. Whether we consider this to be an application of voting in computing or to be a use of computing in voting, the topic lies nicely in the intersection of the two fields of voting theory and computer science. We mention electronic voting in passing here, only as an item that should not be missed in a list of applications combining voting and computing, but we dedicate the following chapter to address this engaging topic in more detail.

5.9 Comments and Bibliographic Notes

This chapter focused on the applications of voting in computing. In other words, the focus here was on concepts imported from voting theory to the realm of computing and on exploiting these ideas to solve problems originating in computer science such as management of autonomous multi-agent systems and ranking methods for search engines. This chapter attempted to exhibit the variety of problems where voting is applicable, it did so by providing a list of some applications in different computing subjects. This list is not exhaustive,

[1] See Chapter 1 of this thesis for definitions of voting rules.

but it gives a general overview. There are, for example, many other applications of voting in problem solving (e.g., [30]), machine learning (e.g., [4, 13, 8, 20]), multimodal interaction (e.g., [31]), and possibly many other topics.

However, the link to the other direction exists and is marked by rich research endeavors. In this direction, computer science concepts and techniques are used to further study voting and to provide fine analyses and practical representations of voting procedures. This scheme as a whole provides new tools, criteria and perspective for describing, evaluating and analyzing voting systems respectively. We can think of most of the work discussed in previous chapters as efforts in this direction. The relative proportions of discussions on these directions is explained by the fact that the thesis is dedicated to the *computational aspects* of voting (and not to voting applications in computing).

The confluence of these two disciplines has enriched interdisciplinary research involving a colorful clique of fields such as decision theory, game theory, mathematical modeling, computational logic, combinatorial auctions and welfare economics.

In the next chapter, we discuss in some detail a topic that lies nicely in the intersection of voting theory and computing, and links both fields in both aforementioned directions: Electronic Voting.

Chapter 5 Bibliography

[1] Ahamad, M., Ammar, M., and Cheung, S. Y. Multidimensional voting. *ACM Transactions on Database Systems 9*, 4 (1991), 399–431.

[2] Altman, A., and Tennenholtz, M. On the axiomatic foundations of ranking systems. In *IJCAI' 05: Proceedings of the 19th International Joint Conference on Artificial Intelligence* (Edinburgh, Scotland, UK, 2005), Professional Book Center, pp. 917–922.

[3] Altman, A., and Tennenholtz, M. Ranking systems: the PageRank axioms. In *EC '05: Proceedings of the 6th ACM Conference on Electronic Commerce* (New York, NY, USA, 2005), ACM Press, pp. 1–8.

[4] Bauer, E., and Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning 36*, 1-2 (1999), 105–139.

[5] Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. Rank aggregation methods for the Web. In *WWW '01: Proceedings of the 10th International Conference on World Wide Web* (New York, NY, USA, 2001), ACM Press, pp. 613–622.

[6] Ephrati, E., and Rosenschein, J. The Clarke tax as a consensus mechanism among automated agents. In *AAAI' 91: Proceedings of the 9th National Conference on Artificial Intelligence* (San Jose, California, 1991), pp. 173–178.

[7] Ephrati, E., and Rosenschein, J. Multi-agent planning as a dynamic search for social consensus. In *IJCAI' 93: Proceedings of the 13th International Joint Conference on Artificial Intelligence* (Chambery, France, 1993), pp. 423–429.

[8] Evgeniou, T., Pontil, M., and Elisseeff, A. Leave one out error, stability, and generalization of voting combinations of classifiers. *Machine Learning 55*, 1 (2004), 71–97.

[9] Fagin, R., Kumar, R., and Sivakumar, D. Efficient similarity search and classification via rank aggregation. In *SIGMOD '03: Proceedings of the ACM SIGMOD (Special Interest Group on Management of Data) International Conference on Management of Data* (New York, NY, USA, 2003), ACM Press, pp. 301–312.

[10] Garcia-Molina, H. Elections in a distributed computing system. *IEEE Transactions on Computers 31*, 1 (1982), 48–59.

[11] Ghosh, S., Mundhe, M., Hernandez, K., and Sen, S. Voting for movies: the anatomy of a recommender system. In *AGENTS '99: Proceedings of the 3rd Annual Conference on Autonomous Agents* (New York, NY, USA, 1999), ACM Press, pp. 434–435.

[12] Gifford, D. Weighted voting for replicated data. In *SOSP '79: Proceedings of the 7th ACM Symposium on Operating Systems Principles* (New York, NY, USA, 1979), ACM Press, pp. 150–162.

[13] Halteren, H. v., Zavrel, J., and Daelemans, W. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics 27*, 2 (2001), 199–229.

[14] Hemaspaandra, E., Hemaspaandra, L., and Rothe, J. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM 44*, 6 (1997), 806–825.

[15] Hemaspaandra, E., Spakowski, H., and Vogel, J. The complexity of Kemeny elections. *Theoretical Computer Science 349*, 3 (2005), 382–391.

[16] Jajodia, S., and Mutchler, D. Enhancements to the voting algorithm. In *VLDB '87: Proceedings of the 13th International Conference on Very Large Databases* (San Francisco, CA, USA, 1987), Morgan Kaufmann Publishers Inc., pp. 399–406.

[17] Jajodia, S., and Mutchler, D. Dynamic voting algorithms for maintaining the consistency of a replicated database. *ACM Transactions on Database Systems 15*, 2 (1990), 230–280.

[18] Konczak, K. Voting theory in answer set programming. In *WLP' 06: Proceedings of the 20th Workshop on Logic Programming* (Vienna, Austria, 2006), M. Fink, H. Tompits, and S. Woltran, Eds., no. INFSYS RR-1843-06-02 in Technical Report Series, Technische Universität Wien, pp. 45–53.

[19] Konczak, K., and Lang, J. Voting procedures with incomplete preferences. In *Proceedings of Multidisciplinary IJCAI'05 Workshop on Advances in Preference Handling* (Edinburg, Scotland, UK, 2005).

[20] Kotsiantis, S., and Pintelas, P. Local voting of weak classifiers. *International Journal of Knowledge-Based and Intelligent Engineering Systems 9*, 3 (2005), 239–248.

[21] Oflazer, K., and Tür, G. Morphological disambiguation by voting constraints. In *EACL' 97: Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics* (Morristown, NJ, USA, 1997), Association for Computational Linguistics, pp. 222–229.

[22] Pennock, D., Horvitz, E., and Giles, L. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *AAAI/IAAI' 00: Proceedings of the 17th National Conference on Artificial Intelligence*

and 12th Conference on Innovative Applications of Artificial Intelligence (2000), AAAI Press / The MIT Press, pp. 729–734.

[23] Qin, M., and Zimmermann, R. An energy-efficient voting-based clustering algorithm for sensor networks. In *SNPD-SAWN '05: Proceedings of the 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and 1st ACIS International Workshop on Self-Assembling Wireless Networks* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 444–451.

[24] Rothe, J., Spakowski, H., and Vogel, J. Exact complexity of the winner problem for Young elections. *Theory of Computing Systems 36*, 4 (2003), 375–386.

[25] Sigletos, G., Paliouras, G., Spyropoulos, C., and Hatzopoulos, M. Combining information extraction systems using voting and stacked generalization. *Journal of Machine Learning Research 6* (2005), 1751–1782.

[26] Sirivianos, M., Westhoff, D., Armknecht, F., and Girao, J. Non-manipulable aggregator node election protocols for wireless sensor networks. In *ICST WiOpt' 07: Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks* (2007).

[27] Tang, J. Voting class—an approach to achieving high availability for replicated data. In *DPDS '90: Proceedings of the 2nd International Symposium on Databases in Parallel and Distributed Systems* (New York, NY, USA, 1990), ACM Press, pp. 146–156.

[28] Thomas, R. A solution to the concurrency control problem for multiple copy databases. In *COMPCON '78: Proceedings of the IEEE Computer Society International Conference* (New York, New York, USA, 1978), IEEE Computer Society, pp. 56–62.

[29] Thomas, R. A majority consensus approach to concurrency control for multiple copy databases. *ACM Transactions on Database Systems 4*, 2 (1979), 180–209.

[30] Tikk, D., Kardkovács, Z., and Szidarovszky, F. Voting with a parameterized veto strategy: solving the KDD Cup 2006 problem by means of a classifier committee. *SIGKDD (ACM Special Interest Group on Knowledge Discovery and Data Mining) Explorations 8*, 2 (2006), 53–62.

[31] Zhang, L.-G., Chen, X., Wang, C., and Gao, W. Hierarchical voting classification scheme for improving visual sign language recognition. In *MULTIMEDIA '05: Proceedings of the 13th Annual ACM International Conference on Multimedia* (New York, NY, USA, 2005), ACM Press, pp. 339–342.

Chapter 6

Electronic Voting

The people who cast the votes don't decide an election, the people who count the votes do.

Joseph Stalin

In Fairfax, Virginia (2003), testing ordered by a judge revealed that several voting machines subtracted one in every hundred votes for the candidate who lost her seat on the School Board.

Cited by **VerifiedVoter.org**[1]

6.0 Prologue

In previous chapters (up to Chapter 5), we have discussed voting mostly from a theoretical perspective. In this chapter, we discuss a large scale application of voting: electronic voting. This discussion will place concepts in a clear practical perspective. For example, we have defined an election to consist of a set C of candidates and a set V of voters. Here, an election includes many other factors and entities. In fact, in the context of electronic voting, most of the work in an election is done before votes are cast, unlike the previous discussions where the analysis focused on phenomena that take place when reporting and aggregating the votes.

This chapter is not intended to survey topics in electronic voting. This is a very broad multifaceted topic that has been extensively studied. The cryptographic aspects alone correspond to a stupendous number of studies (see for example the bibliography in [22]). These aspects are presented in only informal elucidation.

This chapter, however, is dedicated to a limited discussion of electronic voting since this is a very important topic in the computing-voting literature, and its relevance to the topic of this survey is unquestionable. Indeed, the term "voting system" in computing circles stands most frequently for electronic voting, it is relatively recent that the term is also used to stand for voting rules analyzed theoretically.

[1] `http://www.verifiedvoting.org/downloads/resources/hr2239_volunteers/Introduction-5-pages.htm`

In the remaining sections we define *What is Electronic Voting* in more detail, discuss *Voting technology*, catch a glimpse of the *History of Electronic Voting*, state proposed *Criteria for e-Voting Systems*, point out main *Electronic Voting Debates*, present some *Electronic Voting Proposals*, and conclude with *Comments and Bibliographic Notes*.

6.1 What is Electronic Voting?

lectronic voting (e-voting) refers to the use of computers or computerized equipment to cast votes in an election. The term electronic voting is sometimes used to refer to Internet voting (where voters communicate their votes through Internet connection). Electronic systems can be used to register voters, tally ballots (count the votes), and record votes [15].

There are different possible implementations of electronic voting. The literature distinguishes the following three:

- Voting at supervised poll-sites using electronic equipment.
- Voting at electronic kiosks available in voter-convenient places (such as public libraries or shopping centers). The kiosks may be supervised by officials, volunteers or even surveillance cameras.
- Remote voting (from home, work, etc.) using the voter's equipment and the Internet.

Many concerns are associated with electronic voting. For example, the Caltech/MIT Voting Technology Project found that out of the five types of voting machines, hand-counted papers, mechanical lever machines, punch card ballots, optically scanned paper, and electronic voting machines, electronic voting machines, despite the accuracy and reliability appeals for which they were introduced, have the second highest rate of unmarked, uncounted, and damaged ballots in governor, senate, and presidential elections in the past decade [14, 43]. Another example, when the implications of changing only a single vote per machine were investigated [20], it turned out that this seemingly insignificant change can swing the election result. Such a major diversion from accurate results is not likely with occasional errors even when manually counting ballots. Researchers conclude that electronic voting machines can possibly make large scale fraud a simple task of changing the source code of one software system.

However, there are numerous arguments for supporting electronic voting given a clear set of criteria and guidelines. One of the important factors to consider when supporting (or opposing) an electronic voting argument is the voting technology used. This is the topic of the next section.

6.2 Voting Technology

Voting technologies vary greatly and include technologies such as punch cards, optical scanners (that use optical scanners to scan ballots), machine readable ballot systems, direct-recording e-voting machines that input votes via some input device (such as a keyboard or touch screen) and immediately add the vote to a running count. Other means for transmitting the vote such as telephone lines, private networks, or Internet are also considered electronic voting technologies. For a comparative study on the usability of these technologies, see [8].

Today, advanced encryption technologies, signature schemes, identification and authentication schemes, and secret sharing schemes are among the leading topics addressed to improve the security and reliability of electronic voting.

In addition to security and reliability objectives, the use of electronic technologies for voting aims at obtaining accurate results, reducing costs, speeding up voting and tallying processes, and facilitating the voting process and increasing voter participation.

6.3 History of Electronic Voting

Although the application of e-voting is relatively recent, early proposals were made to realize this form of voting. Some of these proposals date back to the 19th century. For example, on June 1st, 1869, Thomas Edison received U.S. Patent 90,646 for an "Electric Vote-Recorder" that was designed for use in Congress [25]. Interestingly, this was Edison's first patented invention. However, it remained a proposal and was not deployed.

In 1955, Froom described a situation where individuals in a groups would communicate using some "technical devices" that would also assist in aggregating the opinions of the group members. In 1970, Henderson suggested "new ways of improving communication channels to inform the voter, and machinery to channel his or her participation." Perhaps the first computer that supported voting system was the Delphi panel that Turoff implemented in 1970. This developed into what is now known as "computer conferencing."

The idea of electronic voting surfaced again in 1971. Fuller used the term "electrified voting" in a theoretical context to describe a system that would realize democracy. The seventies witnessed rich discussions and interest in electronic voting. In 1971 also, Ohlin described a system where homes would be connected to huge databases via terminals, and where citizens could participate in social decision making [35].

At the turn of the century, the number of proposals and active projects exploded dramatically and different computerized voting equipment was in use. During that period, several concerns regarding the reliability and security of such equipment were raised.

In February 1975, the General Accounting Office of Federal Elections (later known as Federal Election Commission) signed an agreement with the National Bureau of Standards to develop guidelines for ensuring the accuracy and security of the computer-based voting. These guidelines and reported comments from practical use of voting systems led the collaboration of the Federal Election Commission (FEC), and the National Bureau of Standards (now known as the National Institute of Standards and Technology) to conduct a feasibility study on developing technical and operational performance standards for electronic voting systems used in the United States. In 1984, "A Report on the Feasibility of Developing Voluntary Standards for Voting Equipment" was produced [19]. The new guidelines adopted by the U.S. Election Assistance Commission (EAC) in 2005, will take effect in December 2007 replacing the 2002 Voting System Standards. For more on the history of developing e-voting standards see, for example, [6]. The use of electronic tools for voting continued and spread across the United States and other countries.

In the early nineties, the use of electronic voting started in Belgium, it has been used widely since then for municipal and general elections. Many countries followed such as Canada, Brazil, Austria, and India where 380 million voters cast their votes on more than a million machine in the world's largest e-voting experience to date [37].

In the Unites States, in 1996, the Reform Party used Internet voting to select a Presidential candidate. This was the first use of Internet voting in national elections. Few years later, in 2000, Internet voting was used in Alaska (Republican Party Presidential poll), and at a large scale in Arizona (Democratic primary).

After the electronic voting crisis in Florida in 2002, the Caltech/MIT Voting Technology Project [14] emerged in order to develop new voting technologies that preclude e-voting problems like the one which occurred in Florida in 2002. This technology-focused work complements the National Commission on Federal Election Reform (Carter/Ford commission) which focuses on political issues.

In recent years, non-governmental organizations started to conduct elections using electronic voting. Also, educational organizations such as universities began to conduct student government elections using computers connected to the university network or to the Interent (see for a recent example [31]).

Perhaps the most challenging and critical use of e-voting remains to be, however, its use in Presidential elections. Although many countries have introduced electronic voting, such as Australia, Belgium, Brazil, Canada, Estonia, France, Germany, India, Ireland, Italy, the Netherlands, Norway, Romania, Switzerland, United Kingdom and the United States, it remains a controversial issue for many reasons. We will briefly mention some of these issues in the following sections.

6.4 Criteria for e-Voting Systems

The following set of criteria captures the main concerns that must be addressed in the design of an electronic voting system. Most of these, unless cited otherwise, were presented in [21] and frequently restated in the literature:

- Completeness: all valid votes are counted, and counted correctly.
- Soundness: the dishonest voter cannot disrupt the election.
- Robustness: the voting scheme must work properly and produce correct outcomes even if a partial failure of the scheme occurs.
- Privacy: secrecy of each and every vote is maintained. Votes should be anonymous. Voters' privacy must be protected[2] (against coercion for example [3, 27]).
- Usability: in addition to security and accuracy, the voting system must have an interface that facilitates its proper use by the voters. The usability aspect must be

[2] This also implies anonymity of the voting system. Anonymity of a voting scheme is defined in Chapter 1.

taken into consideration when designing a voting system (it must be integrated in the system's architecture), and when installing the system (by training election officials and voters and raising public awareness) [16].

- Unreusability: no voter can vote twice.
- Eligibility: only eligible voters can vote. Moreover, each voter votes once only.
- Fairness: nothing must influence the voting process (other than the preferences of voters based on clear presentation of candidates, e.g., leak of intermediate results while voters are casting votes must be prohibited).
- Verifiability: no one can falsify the result of the election.
- Universal Verifiability: a third independent party can check whether votes are counted correctly or not.
- Receipt-freeness: A voter does not get a receipt that can be used to prove how he voted. This safe-guards against vote selling or voting under coercion [26, 17].

There are also sociological criteria such as deliberative and representative democracy, distribution of roles, voter participation, legal concerns, etc. One very important criterion is consideration of special-needs members of the society, and the availability of voting technology that assists these individuals in voting while protecting their rights of privacy and fairness.

In addition to the technical and sociological criteria, there are concerns related to the interaction of these factors, for example, studies have investigated the effects of voting technologies on voting behavior [36].

6.5 Electronic Voting Debates

"*Three issues: privacy, security, and accuracy are at the heart of the e-Voting debate.*" Phillip J. Windley [43], page 134.

6.5.1 Open Source vs. Closed Source Voting Systems

The arguments favoring open source e-voting systems are numerous and known especially since the issue of source-available software has been discussed for a long time. In the context of electronic voting, open source systems are argued to enjoy improved security and quality since they are open to public scrutiny. In addition to this, open source e-voting systems are free from dependency on suppliers, and from restrictions on features, since an open source voting system can be independently modified by the authority to serve specific purposes when needed. These arguments were presented in more detail in [28]. Examples of worldwide use of open source and closed source e-voting systems—with conclusions encouraging open source—are given in [37].

The arguments against open source e-voting systems basically emphasize the practical limitations of the arguments in favor of open source systems. For example, it is argued that the transparency of open source e-voting systems is really limited since it depends highly on voluntary participation and only a minority of open source e-voting projects attracts contributions from specialists in the field. Moreover, it is argued that even with careful inspection of open source systems, there is no guarantee that the studied system is the one actually used on Election Day. Proving this to voters is a challenging problem especially with the likelihood of last minute fixes and therefore of malicious software intrusion [28]. Perhaps the strongest argument that weakens the interest in open source e-voting is the fact that it does not address the underlying fundamental challenges facing e-voting. For example, open source e-voting does not change the problem of preventing insider attacks, and one can argue that it may facilitate outsider attacks. Also, it does not solve the issue of correctly identifying voters while protecting voter privacy and vote secrecy. It does not address the contextual challenges associated with electronic voting such as usability, systematic workflow, and distribution of administrative tasks.

6.5.2 Vote Verifiability vs. Receipt Freeness

The concept of Voter Verified Paper Audit Trail (VVPAT) or Verified Paper Record (VPR), proposed by Mercuri in her PhD dissertation [32], refers to the use of paper receipts to verify that a vote was recorded as intended by the voter. Proponents of the "Mercuri method," named after its creator, argue that it is intended as an independent verification system to assure voters that their votes were cast and recorded correctly.

Opponents of this method, however, raise the concerns of vote buying, coercion, and compromised voter privacy in case a paper receipt is given to the voter. The voter could use this receipt to prove to a third party that he voted as *requested*. In this case, this form of vote verification is argued to be conducive to vote buying or even to coercion. Hence, receipt-freeness is usually flagged as one of the requirements that a successful voting system should meet [26, 17]. There are concerns related to this form of paper vote-receipt, such as cost and usability but the security, integrity, and privacy concerns remain the most significant.

This debate, however, is likely to disappear since there are new proposals for voting schemes where the voter can verify for himself that his vote was cast and recorded correctly without compromising vote secrecy. Some of these proposals will be mentioned in the proposals section.

6.5.3 Voter Convenience vs. Security

There are two topics under this line of debate: one is the issue of remote voting using the voter's machine and Internet voting, and the other is the ease of using a voting system from a voter's point of view. The latter corresponds to issues such as the friendliness of the interface, the special need adjustments, and other user convenience and usability matters.

In the first topic, issues such as coercion and security of platform are raised. Voting remotely, we cannot make sure that the voter has voted according to his preferences or under coercion by some interested party. Moreover, voters' machines are vulnerable to security attacks through open ports, and therefrom, viruses, worms, Trojan horses, automatically downloadable active content, and many other forms of malicious software. The voter's privacy may also be compromised since a hacker can get access to information stored on the machine, or can even monitor active sessions, among which is the voting session. The other issue with indexplatform security platform security is security concerns related to the Internet itself. Reliability of connection, flooding, denial of service attacks, are all concerns related to the use of Internet that remain an issue even if we solve the problem of the voter machine security. See [41] for a discussion on security considerations for remote electronic voting.

The argument *for* remote Internet voting regards voter convenience and reduced cost of running elections since (a) the voter platform is used, (b) no cost is involved for establishing a poll site or (c) training election officials/volunteers for administrating the voting process at the poll sites. Nevertheless, there seems to be stronger argument *against* the use of remote Internet voting. Specialists argue that even with current advancements of e-voting technology, we cannot securely rely on Internet voting. Unless a revolution in such technologies take place, Internet voting will not be used as a reliable method of voting in the foreseeable future.

The other topic, regarding ease of use and improved interface, also spurred a divergence of opinion. The argument for ease of use is clearly strong and simple. To increase voter participation and turnout, the voting system must be easily usable with an easy-to-understand interface. Moreover, there should be special interfaces designed for voters with special needs such as the disabled (for example the visually impaired).

On the other hand, such friendly interfaces will most likely hide complex implementation. Certifying complicated multi-thousand lines of code is not a trivial task. This places a heavy burden on election officials to certify the system. Ironically, this complexity introduced to improve voter convenience may ultimately discourage voters and undermine their trust in the system.

6.5.4 Diverse vs. Unified Voting Systems

Many appealing arguments can be thought of for using the same system or technology across a region when conducting an election on the regional level. For example, consistency, facilitating training of officials and voters (for it is much easier to train users to use one system than to train different users to use a number of different systems), ease of interfacing and transmission of data.

However, Rivest presented two other arguments supporting the use of different systems by different counties in a state. "Just as a woodland's diverse variety of plants can provide better resistance to pathogens than the farmer single crop, so too can a variety of voting technologies provide resistance to an adversary attack," he commented in [40], page 4.

The other arguments he presented regard gaining experience with new voting systems. Since it takes time to test, use and learn lessons from implementing new systems, it might be wise to let individual counties experiment with new technologies as they emerge. By the time old technologies need to be replaced, there will be reasonable feedback to direct decisions of using new technologies.

6.6 Electronic Voting Proposals

One of the earliest proposals on which many recent proposals were based is Fujioka, Okamoto, and Ohta's proposal of "A Practical Secret Voting Scheme for Large Scale Elections" [21]. They described a system of voters, an administrator, and a counter where the voters communicate their votes to the counter through an anonymous channel. They showed that the system ensures the privacy of the voter even if the administrator and counter conspire. They further proved that their system is fair and is resistant to fraud by either the voters or the administrator. The work was based on earlier proposals of signature schemes [10, 30] and anonymous communication channels [9, 11].

Many other proposals followed [1, 5, 23, 2]. Some of the new proposals were described by specialists in the field as revolutionary or at least paradigm shifting [39]. These are the proposals of Chaum, Ryan, and Schneider [12, 42] and Neff [34]. The proposal of Chaum et al. presented a voting scheme where the voter can verify her vote by getting a receipt without violating vote privacy requirements.

Other proposals focused specially on Internet voting [13]. In [24], Hoffman discusses the result of using Internet voting on democracy. A more optimistic discussion on Internet voting based on Arizona's 2000 Democratic party's primary is given in [33].

Research has also emphasized the importance of integrating the administrative aspects of voting into electronic voting systems [29, 44].

There are three new foci in designing electronic voting systems. One is to use cryptography in the architecture of the system and not only in a limited way for ensuring security of parts of the system, such as communication channels.

The other is to shift the emphasis from certifying voting equipment to certifying voting results, since the ultimate goal after all is to have accurate election results not to merely develop advanced technology.

The third focus is on encrypting the voter's choices and to use that encryption as the ballot cast by the voter. This encrypted ballot is posted on a "public bulletin" with the voter's name. This way, only voters can verify their votes and confirm that their votes were considered.

6.7 Comments and Bibliographic Notes

- A short survey of the state of the art in electronic voting is presented in [7]. For an extensive survey on e-voting technology, refer to [18]. Requirements, design, and implementation details of a special type of electronic voting, remote online voting using various communication devises, are spelled out in [38].

- In his technical report on "Electronic Voting," Rivest comments on the dissimilarity between e-commerce and e-voting [40]. This is reminiscent of Black's much older discussion on the dissimilarity between the domains to which the same theory—of committees and decision making—applies [4]. It seems that despite the apparent similarities between the two domains of voting and commerce, and more generally political science and economics, the two domains are intrinsically different. Here, we recall the quote with which we started the first chapter of this thesis.

> *In seeking private interests, we fail to secure greater collective interests. The narrow rationality of self-interest that can benefit us all in market exchange can also prevent us from succeeding in collective endeavors.*
>
> **Russell Hardin** (Collective Actions)

Chapter 6 Bibliography

[1] Adida, B., and Rivest, R. Scratch & vote: self-contained paper-based cryptographic voting. In *WPES '06: Proceedings of the 5th ACM Workshop on Privacy in Electronic Society* (New York, NY, USA, 2006), ACM Press, pp. 29–40.

[2] Baudron, O., Fouque, P.-A., Pointcheval, D., Stern, J., and Poupard, G. Practical multi-candidate election system. In *PODC '01: Proceedings of the 20th Annual ACM Symposium on Principles of Distributed Computing* (New York, NY, USA, 2001), ACM Press, pp. 274–283.

[3] Benaloh, J., and Tuinstra, D. Receipt-free secret-ballot elections (extended abstract). In *STOC '94: Proceedings of the 26th Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1994), ACM Press, pp. 544–553.

[4] Black, D. On the rationale of group decision-making. *The Journal of Political Economy* (1948).

[5] Brandt, F., and Sandholm, T. Decentralized voting with unconditional privacy. In *AAMAS '05: Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems* (New York, NY, USA, 2005), ACM Press, pp. 357–364.

[6] Brown, A. HAVA and standardization of US voting equipment. Available at `http://evote-mass.org/resources/stdzn.htm`.

[7] Bungale, P., and Sridhar, S. Electronic voting – a survey. Department of Computer Science, The Johns Hopkins University. Available at `http://www.cs.jhu.edu/~rubin/courses/sp03/group-reports/group4/group4_lit-review.pdf`.

[8] Byrne, M., Greene, K., and Everett, S. Usability of voting systems: baseline data for paper, punch cards, and lever machines. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), ACM Press, pp. 171–180.

[9] Chaum, D. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM 24*, 2 (1981), 84–88.

[10] Chaum, D. Showing credentials without identification: Signatures transferred between unconditionally unlinkable pseudonyms. In *EUROCRYPT' 85: Proceedings of the Eurocrypt Conference* (1985), pp. 241–244.

[11] Chaum, D. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology 1*, 1 (1988), 65–75.

[12] Chaum, D., Ryan, P., and Schneider, S. A practical voter-verifiable election scheme. In *ESORICS' 05: 10th European Symposium on Research in Computer Security* (Milan, Italy, 2005), S. D. C. di Vimercati, P. Syverson, and D. Gollmann, Eds., vol. 3679 of *Lecture Notes in Computer Science*, Springer, pp. 118–139.

[13] Chen, Y.-Y., ke Jan, J., and Chen, C.-L. The design of a secure anonymous Internet voting system. *Computers & Security 23*, 4 (2004), 330–337.

[14] Caltech/MIT Voting Technology Project. `http://www.vote.caltech.edu/home.htm`.

[15] Cranor, L. Electronic voting. An entry in the Encyclopedia of Computers and Computer History edited by Raul Rojas, 2001. Available at `http://lorrie.cranor.org/pubs/evoting-encyclopedia.html`.

[16] Crawford, D. Forum. *Communications of the ACM 47*, 12 (2004), 11–13.

[17] Delaune, S., Kremer, S., and Ryan, M. Receipt-freeness: Formal definition and fault attacks (extended abstract). In *FEE' 05: Proceedings of the Workshop Frontiers in Electronic Elections* (Milan, Italy, 2005).

[18] E-voting security study. E-Democracy Consultation, U.K. Cabinet Office. Available at `http://www.ictparliament.org/CDTunisi/ict_compendium/paesi/uk/uk54.pdf`.

[19] Federal Election Commission (FEC). Voting system standards. Available at `http://www.fec.gov/agenda/agendas2001/mtgdoc01-62/overview.pdf`.

[20] Franco, A. D., Petro, A., Shear, E., and Vladimirov, V. Small vote manipulations can swing elections. *Communications of the ACM 47*, 10 (2004), 43–45.

[21] Fujioka, A., Okamoto, T., and Ohta, K. A practical secret voting scheme for large scale elections. In *ASIACRYPT '92: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques* (London, UK, 1993), Springer-Verlag, pp. 244–251.

[22] Greenstadt, R. Electronic voting bibliography. Available at `http://theory.lcs.mit.edu/~cis/voting/greenstadt-voting-bibliography.html`.

[23] Haas, B. Engineering better voting systems. In *DocEng '06: Proceedings of the 2006 ACM Symposium on Document Engineering* (New York, NY, USA, 2006), ACM Press, pp. 56–58.

[24] Hoffman, L. Internet voting: will it spur or corrupt democracy? In *CFP '00: Proceedings of the 10th Conference on Computers, Freedom and Privacy* (New York, NY, USA, 2000), ACM Press, pp. 219–223.

[25] Israel, P. *Edison: A Life of Invention.* Wiley; New Ed edition, 2000.

[26] Jakobsson, M., Sako, K., and Impagliazzo, R. Designated verifier proofs and their applications. In *EuroCrypt '96: Proceedings of the Eurocrypt Conference* (1996), Springer-Verlag, pp. 143–154. Article available at `http://citeseer.ist.psu.edu/jakobsson96designated.html`.

[27] Juels, A., Catalano, D., and Jakobsson, M. Coercion-resistant electronic elections. In *WPES '05: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society* (New York, NY, USA, 2005), ACM Press, pp. 61–70.

[28] Kitcat, J. Source availability and e-voting: an advocate recants. *Communications of the ACM 47*, 10 (2004), 65–67.

[29] Lambrinoudakis, C., Kokolakis, S., Karyda, M., Tsoumas, V., Gritzalis, D., and Katsikas, S. Electronic voting systems: Security implications of the administrative workflow. In *DEXA '03: Proceedings of the 14th International Workshop on Database and Expert Systems Applications* (Washington, DC, USA, 2003), IEEE Computer Society, pp. 467–471.

[30] Lee, C.-C., Yang, W.-P., and Hwang, M.-S. Untraceable blind signature schemes based on discrete logarithm problem. *Fundamenta Informaticae 55*, 3-4 (2003), 307–320.

[31] Maeung, S. Electronic voting system for RIT student government elections. Master's thesis, Department of Computer Science, Rochester Institute of Technology, 2005.

[32] Mercuri, R. *Electronic vote tabulation checks and balances.* PhD thesis, Engineering School of the University of Pennsylvania, 2001.

[33] Mohen, J., and Glidden, J. The case for Internet voting. *Communications of the ACM 44*, 1 (2001), 72–85.

[34] Neff, C. A. A verifiable secret shuffle and its application to e-voting. In *CCS '01: Proceedings of the 8th ACM Conference on Computer and Communications Security* (New York, NY, USA, 2001), ACM Press, pp. 116–125.

[35] Ohlin, T. Swedish magazine Svenska Dagbladet, August 1971.

[36] Oostveen, A.-M., and Besselaar, P. V. D. Trust, identity, and the effects of voting technologies on voting behavior. *Social Science Computer Review 23*, 3 (2005), 304–311.

[37] Penha-Lopes, J. Why use an open source e-voting system? In *ITiCSE '05: Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (New York, NY, USA, 2005), ACM Press, pp. 412–412.

[38] Qadah, G., and Taha, R. Electronic voting systems: Requirements, design, and implementation. *Computer Standards & Interfaces 29*, 3 (2007), 376–386.

[39] Rivest, R. Remarks on electronic voting. Available at `http://theory.lcs.mit.edu/~rivest/publications.html`.

[40] Rivest, R. Electronic voting. Technical report, Laboratory for Computer Science, Massachusetts Institute of Technology, 2004. Available at `http://theory.lcs.mit.edu/~rivest/Rivest-ElectronicVoting.pdf`.

[41] Rubin, A. Security considerations for remote electronic voting. *Communications of the ACM 45*, 12 (2002), 39–44.

[42] Vora, P. David Chaum's voter verification using encrypted paper receipts. Available at `http://www.seas.gwu.edu/~poorvi/Chaum/chaum.pdf`.

[43] Windely, P. e-Voting. Extreme Democracy 13, pp. 132–138. Available at `http://www.extremedemocracy.com/chapters/Chapter\%2011-Windley.pdf`.

[44] Xenakis, A., and Macintosh, A. E-electoral administration: organizational lessons learned from the deployment of e-voting in the UK. In *DGR' 05: Proceedings of the 2005 National Conference on Digital Government Research* (2005), Digital Government Research Center, pp. 191–197.

Subject/Term Index

Author Index

References

[1] Ben Adida and Ronald Rivest. Scratch & vote: self-contained paper-based cryptographic voting. In *WPES '06: Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, pages 29–40, New York, NY, USA, 2006. ACM Press.

[2] Mustaque Ahamad, Mostafa Ammar, and Shun Yan Cheung. Multidimensional voting. *ACM Transactions on Database Systems*, 9(4):399–431, 1991.

[3] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. In *STOC '05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 684–693, New York, NY, USA, 2005. ACM Press.

[4] Alon Altman and Moshe Tennenholtz. On the axiomatic foundations of ranking systems. In *IJCAI' 05: Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 917–922, Edinburgh, Scotland, UK, 2005. Professional Book Center.

[5] Alon Altman and Moshe Tennenholtz. Ranking systems: the PageRank axioms. In *EC '05: Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 1–8, New York, NY, USA, 2005. ACM Press.

[6] Kenneth Arrow. *Social Choice and Justice*, volume 1 of *Collected Papers of Kenneth J. Arrow*. The Belknap Press of Harvard University Press, 1983.

[7] Miguel Ballester and Guillaume Haeringer. A characterization of single-peaked preferences. UFAE and IAE Working Papers 656.06, Unitat de Fonaments de l'Anàlisi Econòmica (UAB) and Institut d'Anàlisi Econòmica (CSIC), 2006. Available at: `http://ideas.repec.org/p/aub/autbar/656.06.html`.

[8] Jeffrey Banks. Sophisticated voting outcomes and agenda control. Working Papers 524, Division of the Humanities and Social Sciences, California Institute of Technology, 1984.

[9] Jean Pierre Barthelemy and Bernard Monjardet. The median procedure in cluster analysis and social choice theory. *Mathematical Social Sciences*, 1(3):235–267, 1981.

[10] John Bartholdi, III and James Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[11] John Bartholdi, III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

[12] John Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16:27–40, 1992.

[13] John Bartholdi, III, C. Tovey, and Michael Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6:227–241, 1989.

[14] John Bartholdi, III, C. Tovey, and Michael Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

[15] John Bartholdi,III and Michael Trick. Stable matching with preferences derived from a psychological model. *Operations Research Letters*, 5(1):165–169, 1986.

[16] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In *PODC '01: Proceedings of the 20th Annual ACM Symposium on Principles of Distributed Computing*, pages 274–283, New York, NY, USA, 2001. ACM Press.

[17] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.

[18] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC '94: Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 544–553, New York, NY, USA, 1994. ACM Press.

[19] Duncan Black. On the rationale of group decision-making. *The Journal of Political Economy*, 56(1):23–34, 1948.

[20] Duncan Black. *The Theory of Committees and Elections*. Cambridge University Press, Cambridge, 1958.

[21] Julian Blau. A direct proof of Arrow's theorem. *Econometrica*, 40(1):61–67, 1972.

[22] Avrim Blum, Jeffrey Jackson, Tuomas Sandholm, and Martin Zinkevich. Preference elicitation and query learning. *Journal of Machine Learning Research*, 5:649–667, 2004.

[23] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. Preference-based constrained optimization with CP-nets. *Computational Intelligence*, 20(2):137–157, 2004.

[24] Craig Boutilier. Toward a logic for qualitative decision theory. In Jon Doyle, Erik Sandewall, and Pietro Torasso, editors, *KR' 94: Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, pages 75–86, 1994.

[25] Craig Boutilier, Ronen Brafman, Christopher Geib, and David Poole. A constraint-based approach to preference elicitation and decision making. In Jon Doyle and Richmond H. Thomason, editors, *Working Papers of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, pages 19–28, Menlo Park, CA, USA, 1997. American Association for Artificial Intelligence.

[26] Steven Brams and Peter Fishburn. Voting procedures. In K. J. Arrow, A. K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1 of *Handbook of Social Choice and Welfare*, chapter 4, pages 173–236. Elsevier, 2002.

[27] Felix Brandt and Tuomas Sandholm. Decentralized voting with unconditional privacy. In *AAMAS '05: Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 357–364, New York, NY, USA, 2005. ACM Press.

[28] Eric Brelsford. Approximation and elections. Master's thesis, Computer Science Department, Rochester Institute of Technology, 2007.

[29] Gerhard Brewka. Logic programming with ordered disjunction. In *AAAI' 02: 18th National Conference on Artificial Intelligence*, pages 100–105, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[30] Alex Brown. HAVA and standardization of US voting equipment. Available at `http://evote-mass.org/resources/stdzn.htm`.

[31] Scott Buffett and Bruce Spencer. Learning opponents' preferences in multi-object automated negotiation. In *EC' 05: Proceedings of the 7th International Conference on Electronic Commerce*, pages 300–305, New York, NY, USA, 2005. ACM Press.

[32] Prashanth Bungale and Swaroop Sridhar. Electronic voting – a survey. Department of Computer Science, The Johns Hopkins University. Available at `http://www.cs.jhu.edu/~rubin/courses/sp03/group-reports/group4/group4_lit-review.pdf`.

[33] Michael Byrne, Kristen Greene, and Sarah Everett. Usability of voting systems: baseline data for paper, punch cards, and lever machines. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 171–180, New York, NY, USA, 2007. ACM Press.

[34] Jr. C. D. Mote. Report of the national workshop on Internet voting: issues and research agenda. In *Proceedings of the 2002 Annual National Conference on Digital Government Research*, pages 1–59. Digital Government Research Center, 2002.

[35] National commission on federal election reform. `http://www.reformelections.org/ncfer.asp`.

[36] John Chamberlin and Paul Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review*, 77(4):718–733, 1983.

[37] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[38] David Chaum. Showing credentials without identification: Signatures transferred between unconditionally unlinkable pseudonyms. In *EUROCRYPT' 85: Proceedings of the Eurocrypt Conference*, pages 241–244, 1985.

[39] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[40] David Chaum, Peter Ryan, and Steve Schneider. A practical voter-verifiable election scheme. In di Vimercati et al. [67], pages 118–139.

[41] Li Chen and Pearl Pu. Survey of preference elicitation methods. Technical report, Swiss Federal Institute of Technology in Lausanne, Lausanne, Switzerland, 2004.

[42] Yu-Yi Chen, Jinn ke Jan, and Chin-Ling Chen. The design of a secure anonymous Internet voting system. *Computers & Security*, 23(4):330–337, 2004.

[43] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. A short introduction to computational social choice. In *SOFSEM' 07: Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science*, volume 4362 of *LNCS*. Springer-Verlag, 2007. Available at `http://staff.science.uva.nl/~ulle/pubs/files/ChevaleyreEtAlSOFSEM2007.pdf`.

[44] Caltech/MIT Voting Technology Project. `http://www.vote.caltech.edu/home.htm`.

[45] William Cohen, Robert Schapire, and Yoram Singer. Learning to order things. In *NIPS '97: Proceedings of the 10th Conference on Advances in Neural Information Processing Systems*, pages 451–457, Cambridge, MA, USA, 1998. MIT Press.

[46] Tom Coleman and Vanessa Teague. On the complexity of manipulating elections. In Joachim Gudmundsson and Barry Jay, editors, *CATS' 07: Proceedings of the 13th Computing Conference: The Australasian Theory Symposium*, volume 65 of *CRPIT*, pages 25–33, Ballarat, Australia, 2007. ACS.

[47] Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions. In *EC' 01: Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 256–259, New York, NY, USA, 2001. ACM Press.

[48] Vincent Conitzer. *Computational Aspects of Preference Aggregation.* PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA, 2006.

[49] Vincent Conitzer. Computing Slater rankings using similarities among candidates. In *AAAI' 06: Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference*, Boston, MA, USA, 2006. AAAI Press.

[50] Vincent Conitzer. Eliciting single-peaked preferences using comparison queries. In *AAMAS' 07: Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems*, pages 408–415, Honolulu, HI, USA, 2007.

[51] Vincent Conitzer, Andrew Davenport, and Jayant Kalagnanam. Improved bounds for computing Kemeny rankings. In *AAAI' 06: Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference*, Boston, MA, USA, 2006. AAAI Press.

[52] Vincent Conitzer, Jérôme Lang, and Tuomas Sandholm. How many candidates are needed to make elections hard to manipulate? In *TARK '03: Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 201–214, New York, NY, USA, 2003. ACM Press.

[53] Vincent Conitzer and Tuomas Sandholm. Complexity of manipulating elections with few candidates. In *AAAI' 02: Proceedings of the 18th National Conference on Artificial Intelligence*, pages 314–319, Menlo Park, CA, USA, 2002. AAAI Press.

[54] Vincent Conitzer and Tuomas Sandholm. Vote elicitation: Complexity and strategy-proofness. In *AAAI' 02: Proceedings of the 18th National Conference on Artificial Intelligence*, pages 392–397, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[55] Vincent Conitzer and Tuomas Sandholm. Universal voting protocol tweaks to make manipulation hard. In *IJCAI' 03: Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 781–788, Acapulco, Mexico, 2003.

[56] Vincent Conitzer and Tuomas Sandholm. Common voting rules as maximum likelihood estimators. In *UAI' 05: Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence*, pages 145–152, Edinburgh, Scotland, UK, 2005.

[57] Vincent Conitzer and Tuomas Sandholm. Communication complexity of common voting rules. In *EC ' 05: Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 78–87, New York, NY, USA, 2005. ACM Press.

[58] Vincent Conitzer and Tuomas Sandholm. Communication complexity of common voting rules. In *EC '05: Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 78–87, New York, NY, USA, 2005. ACM Press.

[59] Vincent Conitzer and Tuomas Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *AAAI' 06: Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Conference on Innovative Applications of Artificial Intelligence*, Boston, MA, USA, 2006. AAAI Press.

[60] Vincent Conitzer, Tuomas Sandholm, and Jérôme Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):14, 2007.

[61] Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms.* MIT Press, Cambridge, MA, USA, 2001.

[62] Sylvie Coste-Marquis, Jérôme Lang, Paolo Liberatore, and Pierre Marquis. Expressive power and succinctness of propositional languages for preference representation. In *KR' 04: Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning*, Whistler, Canada, 2004.

[63] Lorrie Cranor. Electronic voting. An entry in the Encyclopedia of Computers and Computer History edited by Raul Rojas, 2001. Available at `http://lorrie.cranor.org/pubs/evoting-encyclopedia.html`.

[64] Diane Crawford. Forum. *Communications of the ACM*, 47(12):11–13, 2004.

[65] Andrew Davenport and Jayant Kalagnanam. A computational study of the Kemeny rule for preference aggregation. In *AAAI' 04: Proceedings of the 19th National Conference on Artificial Intelligence and the 16th Conference on Innovative Applications of Artificial Intelligence*, pages 697–702, San Jose, CA, USA, 2004. AAAI Press / The MIT Press.

[66] Stéphanie Delaune, Steve Kremer, and Mark Ryan. Receipt-freeness: Formal definition and fault attacks (extended abstract). In *FEE' 05: Proceedings of the Workshop Frontiers in Electronic Elections*, Milan, Italy, 2005.

[67] Sabrina De Capitani di Vimercati, Paul Syverson, and Dieter Gollmann, editors. *ESORICS' 05: 10th European Symposium on Research in Computer Security*, volume 3679 of *Lecture Notes in Computer Science*, Milan, Italy, 2005. Springer.

[68] Persi Diaconis and R. Graham. Spearman's footrule as a measure of disarray. Technical Report ADA030697, California Department Statistics, Stanford University, 1976.

[69] Didier Dubois, Hélène Fargier, and Patrice Perny. On the limitations of ordinal approaches to decision-making. In *KR' 02: Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning*, pages 133–146, Toulouse, France, 2002.

[70] Didier Dubois, Hélène Fargier, Patrice Perny, and Henri Prade. On the limitation of ordinal approaches to decision making. In *KR' 02: Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning*, pages 133–146, Toulouse, France, 2002.

[71] Michael Dummett and Robin Farquharson. Stability in voting. *Econometrica*, 29(1):33–43, 1961.

[72] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *World Wide Web*, pages 613–622, 2001.

[73] Edith Elkind and Helger Lipmaa. Hybrid voting protocols and hardness of manipulation. In Xiaotie Deng and Ding-Zhu Du, editors, *ISAAC' 05: Proceedings of the 16th International Symposium on Algorithms and Computation*, volume 3827 of *Lecture Notes in Computer Science*, pages 206–215, Sanya, Hainan, China, 2005. Springer.

[74] Edith Elkind and Helger Lipmaa. Small coalitions cannot manipulate voting. In Andrew Patrick and Moti Yung, editors, *FC' 05: Proceedings of the 9th International Conference on Financial Cryptography and Data Security*, volume 3570 of *Lecture Notes in Computer Science*, pages 285–297. Springer, 2005.

[75] Eithan Ephrati and Jeffrey Rosenschein. The Clarke tax as a consensus mechanism among automated agents. In *AAAI' 91: Proceedings of the 9th National Conference on Artificial Intelligence*, pages 173–178, San Jose, California, 1991.

[76] Eithan Ephrati and Jeffrey Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *IJCAI' 93: Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 423–429, Chambery, France, 1993.

[77] Theodoros Evgeniou, Massimiliano Pontil, and André Elisseeff. Leave one out error, stability, and generalization of voting combinations of classifiers. *Machine Learning*, 55(1):71–97, 2004.

[78] E-voting security study. E-Democracy Consultation, U.K. Cabinet Office. Available at `http://www.ictparliament.org/CDTunisi/ict_compendium/paesi/uk/uk54.pdf`.

[79] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *SIGMOD '03: Proceedings of the ACM SIGMOD (Special Interest Group on Management of Data) International Conference on Management of Data*, pages 301–312, New York, NY, USA, 2003. ACM Press.

[80] Piotr Faliszewski, Edith Hemaspaandra, and Lane Hemaspaandra. The complexity of bribery in elections. In *AAAI' 06: Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Conference on Innovative Applications of Artificial Intelligence*, Boston, MA, USA, 2006. AAAI Press.

[81] Piotr Faliszewski, Edith Hemaspaandra, and Lane Hemaspaandra. How hard is bribery in elections? Technical Report TR-895, Department of Computer Science, University of Rochester, 2006.

[82] Piotr Faliszewski, Edith Hemaspaandra, Lane Hemaspaandra, and Jörg Rothe. Llull and Copeland voting broadly resist bribery and control. In *AAAI' 07: Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 724–730, Vancouver, Canada, 2007.

[83] Hélène Fargier, Jérôme Lang, and Pierre Marquis. Propositional logic and one-stage decision making. In *KR' 00: Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning*, pages 445–456, Breckenridge, CO, USA, 2000.

[84] Federal Election Commission (FEC). Voting system standards. Available at `http://www.fec.gov/agenda/agendas2001/mtgdoc01-62/overview.pdf`.

[85] Dieter Fensel, Fausto Giunchiglia, Deborah McGuinness, and Mary-Anne Williams, editors. *8th International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 2002.

[86] Peter Fishburn. Paradoxes of voting. *The American Political Science Review*, 68(2):537–546, 1974.

[87] Peter Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.

[88] Peter Fishburn. Discrete mathematics in voting and group choice. *SIAM Journal on Algebraic and Discrete Methods*, 5(2):263–275, 1984.

[89] Anthony Di Franco, Andrew Petro, Emmett Shear, and Vladimir Vladimirov. Small vote manipulations can swing elections. *Communications of the ACM*, 47(10):43–45, 2004.

[90] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *ASIACRYPT '92: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, pages 244–251, London, UK, 1993. Springer-Verlag.

[91] Wulf Gaertner. *A Primer in Social Choice Theory*. Oxford University Press, Oxford University Press Inc., first edition, 2006.

[92] Hector Garcia-Molina. Elections in a distributed computing system. *IEEE Transactions on Computers*, 31(1):48–59, 1982.

[93] Michael Garey and David Johnson. *Computers and Intractability; A Guide to the Theory of* NP*-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[94] Sumit Ghosh, Manisha Mundhe, Karina Hernandez, and Sandip Sen. Voting for movies: the anatomy of a recommender system. In *AGENTS '99: Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 434–435, New York, NY, USA, 1999. ACM Press.

[95] Alan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(1):587–601, 1973.

[96] David Gifford. Weighted voting for replicated data. In *SOSP '79: Proceedings of the 7th ACM Symposium on Operating Systems Principles*, pages 150–162, New York, NY, USA, 1979. ACM Press.

[97] Michel Grabisch. On preference representation on an ordinal scale. In *ECSQARU' 01: Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 18–28, London, UK, 2001. Springer-Verlag.

[98] James Green-Armytage. A survey of basic voting methods. `http://fc.antioch.edu/~james_green-armytage/vm/survey.htm`.

[99] Rachel Greenstadt. Electronic voting bibliography. Available at `http://theory.lcs.mit.edu/~cis/voting/greenstadt-voting-bibliography.html`.

[100] Bertrand Haas. Engineering better voting systems. In *DocEng '06: Proceedings of the 2006 ACM Symposium on Document Engineering*, pages 56–58, New York, NY, USA, 2006. ACM Press.

[101] Gunter Hagele and Friedrich Pukelsheim. The electoral writings of Ramon Llull. *Studia Lulliana*, 41(97):3–38, 2001.

[102] Hans van Halteren, Jakub Zavrel, and Walter Daelemans. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics*, 27(2):199–229, 2001.

[103] Edith Hemaspaandra and Lane Hemaspaandra. Computational politics: Electoral systems. In *MFCS '00: Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science*, pages 64–83, London, UK, 2000.

[104] Edith Hemaspaandra and Lane Hemaspaandra. Dichotomy for voting systems. Technical Report TR-861, Department of Computer Science, University of Rochester, 2005.

[105] Edith Hemaspaandra and Lane Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.

[106] Edith Hemaspaandra, Lane Hemaspaandra, and Joerg Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

[107] Edith Hemaspaandra, Lane Hemaspaandra, and Jörg Rothe. Anyone but him: The complexity of precluding an alternative. In *AAAI' 05: Proceedings of the 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference*, pages 95–101, Pittsburgh, PE, USA, 2005. AAAI Press / The MIT Press.

[108] Edith Hemaspaandra, Lane Hemaspaandra, and Jörg Rothe. Hybrid elections broaden complexity-theoretic resistance to control. Technical Report TR-900, Department of Computer Science, University of Rochester, 2006. Conference version appears in *IJCAI' 07: Proceedings of the 20th International Joint Conference on Artificial Intelligence* (Hyderabad, India, 2007).

[109] Edith Hemaspaandra, Lane Hemaspaandra, and Jörg Rothe. Hybrid elections broaden complexity-theoretic resistance to control. In *IJCAI' 07: Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1308–1314, Hyderabad, India, 2007.

[110] Edith Hemaspaandra, Holger Spakowski, and Joerg Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

[111] Lane Hemaspaandra and Mitsunori Ogihara. *The Complexity Theory Companion.* Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[112] Lane Hemaspaandra, Kulathur Rajasethupathy, Prasanna Sethupathy, and Marius Zimand. Power balance and Congressional apportionment algorithms. Technical report, Computer Science Department, University of Rochester, Rochester, NY, USA, 1996.

[113] Paul Herrnson, Benjamin Bederson, Bongshin Lee, Peter Francia, Robert Sherman, Frederick Conrad, Michael Traugott, and Richard Niemi. Early appraisals of electronic voting. *Social Science Computer Review*, 23(3):274–292, 2005.

[114] Lance Hoffman. Internet voting: will it spur or corrupt democracy? In *CFP '00: Proceedings of the 10th Conference on Computers, Freedom and Privacy*, pages 219–223, New York, NY, USA, 2000. ACM Press.

[115] Christopher Homan and Lane Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. In *MFCS' 06: Proceedings of the Mathematical Foundations of Computer Science*, volume 4162 of *Lecture Notes in Computer Science*, pages 528–539, Stará Lesná, Slovakia, 2006. Springer.

[116] John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, second edition, 2001.

[117] Benoît Hudson and Tuomas Sandholm. Effectiveness of preference elicitation in combinatorial auctions. In *AAMAS '02: Revised Papers from the Workshop on Agent Mediated Electronic Commerce on Agent-Mediated Electronic Commerce IV, Designing Mechanisms and Systems*, pages 69–86, London, UK, 2002. Springer-Verlag.

[118] Paul Israel. *Edison: A Life of Invention.* Wiley; New Ed edition, 2000.

[119] Sushil Jajodia and David Mutchler. Enhancements to the voting algorithm. In *VLDB '87: Proceedings of the 13th International Conference on Very Large Databases*, pages 399–406, San Francisco, CA, USA, 1987. Morgan Kaufmann Publishers Inc.

[120] Sushil Jajodia and David Mutchler. Dynamic voting algorithms for maintaining the consistency of a replicated database. *ACM Transactions on Database Systems*, 15(2):230–280, 1990.

[121] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *EuroCrypt '96: Proceedings of the Eurocrypt Conference*, pages 143–154. Springer-Verlag, 1996. Article available at `http://citeseer.ist.psu.edu/jakobsson96designated.html`.

[122] Paul Johnson. *Social Choice: Theory and Research.* Number 123 in Quantitative Applications in the Social Sciences. Sage Publications, Inc., Thousand Oaks, CA: Sage, 1998.

[123] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *WPES '05: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, pages 61–70, New York, NY, USA, 2005. ACM Press.

[124] Leslie Pack Kaelbling and Alessandro Saffiotti, editors. *IJCAI' 05: Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, UK, 2005. Professional Book Center.

[125] John Kemeny. Mathematics without numbers. *Daedalus*, 88:575–591, 1959.

[126] John Kemeny and J. Laurie Snell. *Mathematical Models in the Social Sciences*. Introduction to Higher Mathematics. The MIT Press, 1973. Second MIT press edition.

[127] Jason Kitcat. Source availability and e-voting: an advocate recants. *Communications of the ACM*, 47(10):65–67, 2004.

[128] Kathrin Konczak. Voting theory in answer set programming. In M. Fink, H. Tompits, and S. Woltran, editors, *WLP' 06: Proceedings of the 20th Workshop on Logic Programming*, number INFSYS RR-1843-06-02 in Technical Report Series, pages 45–53, Vienna, Austria, 2006. Technische Universität Wien.

[129] Kathrin Konczak and Jérôme Lang. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI'05 (International Joint Conferences on Artificial Intelligence) Workshop on Advances in Preference Handling*, 2005.

[130] S. Kotsiantis and P. Pintelas. Local voting of weak classifiers. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 9(3):239–248, 2005.

[131] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1996.

[132] Celine Lafage and Jérôme Lang. Logical representation of preferences for group decision making. In *KR' 00: Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning*, pages 457–468, Breckenridge, CO, USA, 2000.

[133] Celine Lafage and Jérôme Lang. Propositional distances and preference representation. In *ECSQARU '01: Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 48–59, London, UK, 2001. Springer-Verlag.

[134] Sebastien Lahaie and David Parkes. Applying learning algorithms to preference elicitation. In *EC' 04: Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 180–188, New York, NY, USA, 2004. ACM Press.

[135] Costas Lambrinoudakis, Spyros Kokolakis, Maria Karyda, Vasilis Tsoumas, Dimitris Gritzalis, and Sokratis Katsikas. Electronic voting systems: Security implications of the administrative workflow. In *DEXA '03: Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, pages 467–471, Washington, DC, USA, 2003. IEEE Computer Society.

[136] Jérôme Lang. Some representation and computational issues in social choice. In *ECSQARU' 05: Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 3571 of *Lecture Notes in Computer Science*, pages 15–26. Springer, 2005.

[137] Jérôme Lang, Maria Pini, Francesca Rossi, Kristen Venable, and Toby Walsh. Winner determination in sequential majority voting. In *IJCAI' 07: Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1372–1377, Hyderabad, India, 2007.

[138] J.-F. Laslier. *Tournament Solutions and Majority Voting*. Springer Verlag, Berlin, 1997.

[139] Cheng-Chi Lee, Wei-Pang Yang, and Min-Shiang Hwang. Untraceable blind signature schemes based on discrete logarithm problem. *Fundamenta Informaticae*, 55(3-4):307–320, 2003.

[140] Peter Linz. *An Introduction to Formal Languages and Automata*. Jones and Bartlett, Sudbury, MA, USA, fourth edition, 2006.

[141] Sungho Maeung. Electronic voting system for RIT student government elections. Master's thesis, Department of Computer Science, Rochester Institute of Technology, 2005.

[142] Marquis de Condorcet. Essai sur l'application de l'analyse à la probabilité des decisions rendues à la pluralite des voix, 1785.

[143] John Martin. *Introduction to Languages and the Theory of Computation*. McGraw-Hill, third edition, 2002.

[144] Tomomi Matsui and Yasuko Matsui. A survey of algorithms for calculating power indices of weighted majority games. *Journal of the Operations Research Society of Japan*, 43(1):71–86, 2000.

[145] Yasuko Matsui and Tomomi Matsui. NP-completeness for calculating power indices of weighted majority games. *Theoretical Computer Science*, 263(1,2):305–310, 2001.

[146] Stefan Maus, Hans Peters, and Ton Storcken. Anonymous voting and minimal manipulability. Research Memoranda 009, Maastricht: METEOR, Maastricht Research School of Economics of Technology and Organization, 2005. Available at `http://ideas.repec.org/p/dgr/umamet/2005009.html`.

[147] Rebecca Mercuri. *Electronic vote tabulation checks and balances*. PhD thesis, Engineering School of the University of Pennsylvania, 2001.

[148] Joe Mohen and Julia Glidden. The case for Internet voting. *Communications of the ACM*, 44(1):72–85, 2001.

[149] Burt Monroe. Fully proportional representation. *American Political Science Review*, 89(4):925–940, 1995.

[150] James Moore, William Richmond, and Andrew Whinston. A decision-theoretic approach to information retrieval. *ACM Transactions on Database Systems*, 15(3):311–340, 1990.

[151] Herve Moulin. *Axioms of Cooperative Decision Making*. Econometric Society Monographs. Cambridge University Press, second edition, 2004.

[152] C. Andrew Neff. A verifiable secret shuffle and its application to e-voting. In *CCS '01: Proceedings of the 8th ACM Conference on Computer and Communications Security*, pages 116–125, New York, NY, USA, 2001. ACM Press.

[153] Peter Neumann. Risks to the public in computers and related systems. *SIGSOFT (Special Interest Group on Software Engineering) Software Engineering Notes*, 27(2):5–19, 2002.

[154] Peter Neumann. Risks to the public in computers and related systems. *SIGSOFT (Special Interest Group on Software Engineering) Software Engineering Notes*, 27(3):5–19, 2002.

[155] Richard Niemi and William Riker. The choice of voting systems. *Scientific American*, 234:21–27, 1976.

[156] Hannu Nurmi. Voting procedures: A summary analysis. *British Journal of Political Science*, 13(2):181–208, 1983.

[157] Hannu Nurmi. Voting paradoxes and how to deal with them. *Public Choice*, 107(1,2):194–197, 2001.

[158] Kemal Oflazer and Gökhan Tür. Morphological disambiguation by voting constraints. In *EACL' 97: Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics*, pages 222–229, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

[159] Tomas Ohlin. Swedish magazine Svenska Dagbladet, August 1971.

[160] Anne-Marie Oostveen and Peter Van Den Besselaar. Trust, identity, and the effects of voting technologies on voting behavior. *Social Science Computer Review*, 23(3):304–311, 2005.

[161] Anne-Marie Oostveen and Peter Van Den Besselaar. Trust, identity, and the effects of voting technologies on voting behavior. *Social Science Computer Review*, 23(3):304–311, 2005.

[162] Joana Penha-Lopes. Why use an open source e-voting system? In *ITiCSE '05: Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 412–412, New York, NY, USA, 2005. ACM Press.

[163] David Pennock, Eric Horvitz, and Lee Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In

AAAI/IAAI' 00: Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence, pages 729–734. AAAI Press / The MIT Press, 2000.

[164] Maria Pini, Francesca Rossi, Kristen Venable, and Toby Walsh. Strategic voting when aggregating partially ordered preferences. In *AAMAS '06: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 685–687, New York, NY, USA, 2006. ACM Press.

[165] Maria Pini, Francesca Rossi, Kristen Venable, and Toby Walsh. Incompleteness and incomparability in preference aggregation. In *IJCAI' 07: Proceedings of the 20th International Joint Conferences on Artificial Intelligence*, pages 1464–1469, 2007.

[166] Maria Pini, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Aggregating partially ordered preferences: impossibility and possibility results. In *TARK '05: Proceedings of the 10th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 193–206, Singapore, Singapore, 2005. National University of Singapore.

[167] Charles Plott. Path independence, rationality, and social choice. *Econometrica*, 41(6):1075–1091, 1973.

[168] Charles Plott. Axiomatic social choice theory: An overview and interpretation. *American Journal of Political Science*, 20(3):511–596, 1976.

[169] Geoffrey Pritchard and Arkadii Slinko. On the average minimum size of a manipulating coalition. *Social Choice and Welfare*, 27(2):263–277, 2006. Available at `http://ideas.repec.org/a/spr/sochwe/v27y2006i2p263-277.html`.

[170] Ariel Procaccia and Jeffrey Rosenschein. Junta distributions and the average-case complexity of manipulating elections. In *AAMAS '06: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 497–504, New York, NY, USA, 2006. ACM Press.

[171] Ariel Procaccia, Jeffrey Rosenschein, and Aviv Zohar. Multi-winner elections: Complexity of manipulation, control and winner-determination. In *IJCAI' 07: Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1476–1481, Hyderabad, India, 2007.

[172] Ghassan Qadah and Rani Taha. Electronic voting systems: Requirements, design, and implementation. *Computer Standards & Interfaces*, 29(3):376–386, 2007.

[173] Min Qin and Roger Zimmermann. An energy-efficient voting-based clustering algorithm for sensor networks. In *SNPD-SAWN '05: Proceedings of the 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and 1st ACIS International Workshop on Self-Assembling Wireless Networks*, pages 444–451, Washington, DC, USA, 2005. IEEE Computer Society.

[174] Mario Di Raimondo and Rosario Gennaro. New approaches for deniable authentication. In *CCS '05: Proceedings of the 12th ACM Conference on Computer and Communications Security*, pages 112–121, New York, NY, USA, 2005. ACM Press.

[175] Ronald Rivest. Remarks on electronic voting. Available at `http://theory.lcs.mit.edu/~rivest/publications.html`.

[176] Ronald Rivest. Electronic voting. Technical report, Laboratory for Computer Science, Massachusetts Institute of Technology, 2004. Available at `http://theory.lcs.mit.edu/~rivest/Rivest-ElectronicVoting.pdf`.

[177] Fred Roberts. *Group Decisionmaking*, chapter 7, pages 425–472. Discrete Mathematical Models with Applications to Social, Biological, and Environmental Problems. Prentice-Hall, Englewood Cliffs, NJ, USA, 1976.

[178] Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Aggregating preferences cannot be fair. *Intelligenza Artificiale*, 2(1):30–38, 2005.

[179] Jörg Rothe, Holger Spakowski, and Jörg Vogel. Exact complexity of the winner problem for Young elections. *Theory of Computing Systems*, 36(4):375–386, 2003.

[180] Aviel Rubin. Security considerations for remote electronic voting. *Communications of the ACM*, 45(12):39–44, 2002.

[181] Mark Allen Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(1):187–217, 1974.

[182] Ilya Segal. The communication requirements of of social choice rules and supporting budget sets. Economics Working Papers 0039, Institute for Advanced Study, School of Social Science, Princeton University, 2004. Available at `http://ideas.repec.org/p/ads/wpaper/0039.html`.

[183] R. Shirey. Internet security glossary, 2000.

[184] Georgios Sigletos, Georgios Paliouras, Constantine Spyropoulos, and Michalis Hatzopoulos. Combining information extraction systems using voting and stacked generalization. *Journal of Machine Learning Research*, 6:1751–1782, 2005.

[185] Michael Sirivianos, Dirk Westhoff, Frederik Armknecht, and Joao Girao. Non-manipulable aggregator node election protocols for wireless sensor networks. In *ICST WiOpt' 07: Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2007.

[186] Philip Straffin, Jr. *Topics in the theory of voting*. The UMAP expository monograph series. Birkhauser Verlag AG, 1980.

[187] Alexander Tabarrok. Paradoxes of voting. Available at `http://mason.gmu.edu/~atabarro/paradoxofvoting.pdf`.

[188] Sek-Wah Tan and Judea Pearl. Specification and evaluation of preferences under uncertainty. In J. Doyle, E. Sandewall, and P. Torasso, editors, *KR' 94: Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, pages 530–539, San Francisco, CA, USA, 1994. Kaufmann.

[189] Jian Tang. Voting class—an approach to achieving high availability for replicated data. In *DPDS '90: Proceedings of the 2nd International Symposium on Databases in Parallel and Distributed Systems*, pages 146–156, New York, NY, USA, 1990. ACM Press.

[190] Robert Thomas. A solution to the concurrency control problem for multiple copy databases. In *COMPCON '78: Proceedings of the IEEE Computer Society International Conference*, pages 56–62, New York, New York, USA, 1978. IEEE Computer Society.

[191] Robert Thomas. A majority consensus approach to concurrency control for multiple copy databases. *ACM Transactions on Database Systems*, 4(2):180–209, 1979.

[192] Domonkos Tikk, Zsolt Kardkovács, and Ferenc Szidarovszky. Voting with a parameterized veto strategy: solving the KDD Cup 2006 problem by means of a classifier committee. *SIGKDD (ACM Special Interest Group on Knowledge Discovery and Data Mining) Explorations*, 8(2):53–62, 2006.

[193] Michel Truchon. An extension of the Concordet criterion and Kemeny orders. Cahiers de recherche 9813, Université Laval, Département d'économique, 1998. Available at `http://ideas.repec.org/p/lvl/laeccr/9813.html`.

[194] Takeaki Uno. Efficient computation of power indices for weighted majority games. Technical Report NII-2003-006E, National Institute of Informatics, Japan, 2003.

[195] Arnold Urken. Social choice theory and distributed decision making. In *TC-OA' 88: Proceedings of the ACM SIGOIS and IEEECS Conference on Office Information Systems*, pages 158–168, New York, NY, USA, 1988. ACM Press.

[196] William Vickrey. Utility, strategy, and social decision rules. *The Quarterly Journal of Economics*, 74(4):507–535, 1960.

[197] Poorvi Vora. David Chaum's voter verification using encrypted paper receipts. Available at `http://www.seas.gwu.edu/~poorvi/Chaum/chaum.pdf`.

[198] VerifiedVoting.org. `http://www.verifiedvoting.org`.

[199] Yoshiko Wakabayashi. Aggregation of binary relations: Algorithmic and polyhedral investigations, 1986. Doctoral Dissertation. University of Augsburg.

[200] Ingo Wegener. *Complexity Theory: Exploring the Limits of Efficient Algorithms.* Springer, Berlin, Germany, 2005.

[201] Michael Wellman and Jon Doyle. Preferential semantics for goals. In *AAAI' 91: Proceedings of the 9th National Conference on Artificial Intelligence*, volume 2, pages 698–703, Anaheim, CA, USA, 1991. AAAI Press/MIT Press.

[202] Phillip Windely. e-Voting. Extreme Democracy 13, pp. 132–138. Available at `http://www.extremedemocracy.com/chapters/Chapter\%2011-Windley.pdf`.

[203] Alexandros Xenakis and Ann Macintosh. E-electoral administration: organizational lessons learned from the deployment of e-voting in the UK. In *DGR' 05: Proceedings of the 2005 National Conference on Digital Government Research*, pages 191–197. Digital Government Research Center, 2005.

[204] Andrew Yao. Some complexity questions related to distributive computing (preliminary report). In *STOC '79: Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, pages 209–213, New York, NY, USA, 1979. ACM Press.

[205] H. Young. Optimal voting rules. *Journal of Economic Perspectives*, 9(1):51–64, 1995.

[206] H. Young and A. Levenglick. A consistent extension of Condorcet's election principle. *SIAM Journal on Applied Mathematics*, 35(2):285–300, 1978.

[207] H. P. Young. Extending Condorcet's rule. *Journal of Economic Theory*, 16(2):335–353, 1977.

[208] Liang-Guo Zhang, Xilin Chen, Chunli Wang, and Wen Gao. Hierarchical voting classification scheme for improving visual sign language recognition. In *MULTIMEDIA '05: Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 339–342, New York, NY, USA, 2005. ACM Press.

[209] Martin Zinkevich, Avrim Blum, and Tuomas Sandholm. On polynomial-time preference elicitation with value queries. In *EC' 03: Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 176–185, New York, NY, USA, 2003. ACM Press.